

TEKNILLINEN KORKEAKOULU  
RAKENNUS- JA MAANMITTAUSTEKNIIKANOSASTO

Tuomo Talvitie

MIKROTIETOKONEPOHJAISTEN ASiantuntijajärjestelmien  
RAKENTAMINEN JA KÄYTTÖ TIETEKNIIKASSA

TEKNILLINEN KORKEAKOULU  
RAKENNE- JA YHDYSKUNTATEKNIIKAN  
LAITOSTEN KIRJASTO

Diplomityö, joka on jätetty opinnäytteenä tarkastettavaksi  
diplomi-insinöörin tutkintoa varten Espoossa 23.05.1989

Työn valvoja : vs. apulaisprofessori Pentti Lindgren

Työn ohjaaja : vs. yliassistentti Hannu Hyyppä

## ALKUSANAT

Tämä diplomityö on tehty Teknillisessä Korkeakoulussa Rakennus- ja Maanmittaustekniikan osastolla, tietekniikan laboratoriossa.

Erityiset kiitokset diplomityön ohjaajalle vs. yliassistentti Hannu Hyypälle ja työn valvojalle vs. apulaisprofessorille Pentti Lindgrenille työhöni kohdistuneesta mielenkiinnosta. Kiitokset myös vs. laboratorioinsinööri Jarkko Valtoselle ja myyntineuvottelija Artturi Tarjanteelle hyvistä neuvoista.

Espoossa toukokuun 21. päivänä 1989



Tuomo Talvitie



# SISÄLLYSLUETTELO

## ALKUSANAT

## TIIIVISTELMÄ

## ABSTRACT

|  |    |
|--|----|
| 1. JOHDANTO  | 8  |
| 1.1 Peruskäsitteet   | 9  |
| 1.2 Muita käytettäviä termejä                                  | 10 |
| 1.3 Tietämystekniikan soveltuvuus tietekniikan tarpeisiin      | 12 |
| 2. TIETÄMYKSEN ESITTÄMISTEKNIIKAT                              |    |
| 2.1 Yleistä  | 15 |
| 2.2 Lauselogiikka  | 16 |
| 2.3 Predikaattilogiikka  | 17 |
| 2.4 Produktiotekniikka   | 18 |
| 2.41 Produktiotekniikan ominaisuuksia                          | 20 |
| 2.5 Semanttiset verkot   | 20 |
| 2.51 Semanttisten verkkojen toimintaperiaate                   | 21 |
| 2.52 Semanttisten verkkojen ominaisuuksia                      | 22 |
| 2.6 Kehystekniikka   | 22 |
| 2.61 Proseduraalinen tietämys kehystekniikassa                 | 23 |
| 2.62 Kehystekniikan toimintaperiaate                           | 23 |
| 2.63 Kehystekniikan ominaisuuksia                              | 24 |
| 2.7 Muita tietämyksen esittämistekniikoita                     | 24 |
| 2.8 Esittämistekniikoiden yhdistely                            | 25 |
| 2.9 Tietämyksen esittämisen tasot                              | 25 |
| 3. ASiantuntijajärjestelmän rakentaminen                       |    |
| 3.1 Yleistä  | 26 |
| 3.2 Pienten asiantuntijajärjestelmien kehittäminen             | 28 |
| 3.3 Keskikokoisten asiantuntijajärjestelmien rakentamisvaiheet | 29 |
| 3.31 Esitutkimus   | 29 |
| 3.32 Tehtävän analysointi                                      | 29 |
| 3.33 Prototyypin kehittäminen                                  | 30 |

|   |    |
|---|----|
| 3.34 Järjestelmän kehittäminen  | 31 |
| 3.35 Testaaminen käytännössä  | 31 |
| 3.36 Käyttöönotto   | 32 |
| 3.37 Ylläpito   | 32 |
| 3.4 Tietämyksen hankinta  | 33 |
| <br>4. ASIANTUNTIJAJÄRJESTELMIEN RAKENTAMISEN APUVÄLINEET                                       |    |
| 4.1 Yleistä   | 37 |
| 4.2 Kehittimien luokkajako  | 38 |
| 4.21 Induktiiviset kehittimet   | 38 |
| 4.22 Yksinkertaiset sääntöpohjaiset kehittimet  | 39 |
| 4.23 Rakenteelliset sääntöpohjaiset kehittimet  | 39 |
| 4.24 Hybridikehittimet  | 40 |
| 4.25 Sovellusaluekehittimet   | 41 |
| <br>5. TIETEKNIIKAN ASIANTUNTIJAJÄRJESTELMIÄ  |    |
| 5.1 CHINA   | 42 |
| 5.11 Testitapaus 1  | 44 |
| 5.12 Testitapaus 2  | 47 |
| 5.2 SCEPTRE (Surface Condition Expert for Rehabilitation Planning)                              | 51 |
| 5.3 Asiantuntijajärjestelmä maantien ja rautatien tasoristeyksien turvallisuuden parantamiseksi | 56 |
| 5.4 Muita tietekniikan asiantuntijajärjestelmiä   | 64 |
| 5.41 ROSE   | 64 |
| 5.42 Betonipäällysten arvioija  | 65 |
| 5.43 STREET SMART   | 65 |
| 5.44 OVERDRIVE  | 67 |
| 5.45 INTERSECTION ADVISOR   | 68 |
| 5.46 Tien rakenteen suunnittelun tietämysjärjestelmä  | 69 |
| <br>6. MIKROTIETOKONEPOHJAISEN ASIANTUNTIJAJÄRJESTELMÄKEHITTIMIEN VALINTA                       |    |
| 6.1 Yleistä   | 70 |
| 6.2 Valittavalle kehittimelle asetettavia vaatimuksia   | 71 |
| 6.21 Kehittäjäliityntä  | 71 |
| 6.22 Käyttäjäliityntä   | 71 |
| 6.23 Ohjelmointikieli   | 72 |

|  |     |
|--|-----|
| 6.3 Markkinoilla olevia mikrotietokonepohjaisia<br>asiantuntijajärjestelmäkehittämiä | 73  |
| 6.31 VP-Expert   | 74  |
| 6.32 Xi Plus   | 74  |
| 6.4 Valitun asiantuntijajärjestelmäkehittimen<br>ominaisuuksia                       | 75  |
| 7. MURSKAUSHARJOITUS ASIAANTUNTIJAJÄRJESTELMÄ MURMUR                                 |     |
| 7.1 Yleistä  | 78  |
| 7.2 MURMUR   | 79  |
| 8. JÄRJESTELMÄN ARVIOINTI  |     |
| 8.1 Yleistä  | 85  |
| 8.2 Kehittäjäliityntä  | 87  |
| 8.3 Käyttäjäliityntä   | 87  |
| 8.4 Muuteltavuus   | 88  |
| 8.5 Puutteet   | 89  |
| 9. TULEVAISUUDEN NÄKYMIÄ   |     |
| 9.1 Yleistä  | 91  |
| 9.2 Tietämystekniikan kehittyminen   | 91  |
| 9.3 Tietekniikan sovelluksien lisääntyminen  | 96  |
| 10. PÄÄTELMÄT  | 98  |
| 11. YHTEENVETO   | 101 |
| ENGLISH SUMMARY  | 103 |
| LÄHDELUETTELO  | 105 |



Tekijä ja työn nimi : Tuomo Talvitie

Mikrotietokonepohjaisten asiantuntijajärjestelmien  
rakentaminen ja käyttö tietekniikassa

Päivämäärä : 22.05.1989

Sivumäärä : 107

Osasto :

Professuuri :

Rakennus- ja Maanmittaustekniikka

Tietekniikka

Työn valvoja :

Vs. apulaisprofessori Pentti Lindgren

Työn ohjaaja :

Vs. yliassistentti Hannu Hyyppä

Tämän diplomityön tarkoituksena oli tutkia asiantuntijajärjestelmien soveltuvuutta tietekniikan tarpeisiin. Asiantuntijajärjestelmät ovat tietokoneohjelmia, jotka jäljittelevät inhimillistä päätöksentekoa. Työssä on selvitetty myös asiantuntijajärjestelmien rakentamisvaiheita ja niihin liittyviä ongelmia. Markkinoilla olevia kehittämiä vertailtiin ja valitulla kehittimellä rakennettiin mikrotietokonepohjainen asiantuntijajärjestelmä.

Tietekniikassa on useita potentiaalisia sovelluskohteita asiantuntijajärjestelmien käytölle. Yhdysvalloissa on kehitetty useita erilaisiin suunnittelun ja ylläpidon ongelmiin tarkoitettuja tietekniikan asiantuntijajärjestelmiä.

Murskausharjoitus asiantuntijajärjestelmä rakennettiin Xi Plus-kehittimellä. Järjestelmä perustuu maa- ja kalliorakennustöiden murskausharjoitukseen ja sisältää tiedot harjoituksessa käytettävistä murskaimista. Lähtötietojen perusteella järjestelmä valitsee sopivat murskaimet ja laskee murskaukseen kuluvan ajan sekä murskauksessa syntyvät lajitteet.

Asiantuntijajärjestelmät on havaittu hyviksi työkaluiksi rajattujen ongelmien ratkaisuun, jossa tarvitaan asiantuntemusta. Tietämystekniikka on nopeasti kehittyvä ala ja sen käyttö käytännön ongelmien ratkaisuun tulee yleistymään nopeasti.

Author and name of the thesis : Tuomo Talvitie

The Development and the Use of Microcomputer-based  
Expert Systems in Highway Engineering

Date : May 22, 1989

Number of pages : 107

Department : Faculty of Civil  
Engineering and Surveing

Professorship : Road and  
Railway Engineering

Supervisor :

Assistant professor Pentti Lindgren

Instructor :

Senior assistant Hannu Hyyppä

The purpose of this study was to evaluate the applicability of microcomputer-based expert systems in highway engineering. Expert systems are computer programs that use knowledge and reasoning techniques to solve problems that normally require human expertise. In this study also the development and the problems related to the development of microcomputer-based expert systems are studied. Commercially available expert system shells were evaluated and a prototype expert system was built with the shell that was chosen.

The field of highway engineering is full of ill-structured problems, where human expertise is required. There is therefore great potential in the use of expert systems. In the United States there are several microcomputer-based expert systems for planning and maintenance problems in highway engineering.

A demonstration prototype expert system was built to assist students in planning an aggregate crushing plant. The system was developed with Xi Plus. The expert system chooses the right crushers and calculates the time of the crushing and the amount of each fraction that is being crushed.

Expert systems are found to be efficient tools for small, ill-structured problems that require human expertise. Knowledge engineering is a rapidly growing field and its practical applications will become more popular in the near future.



## 1. JOHDANTO

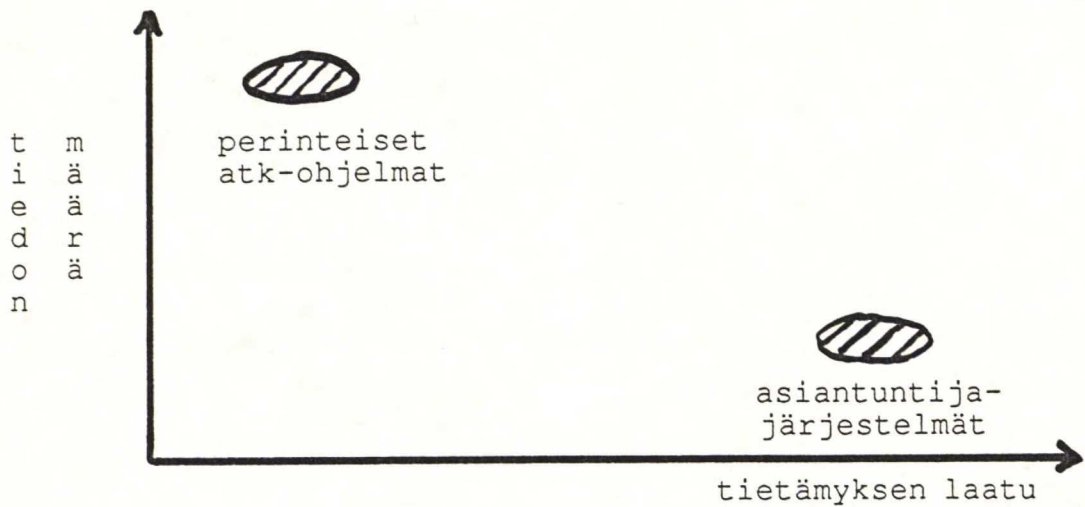
Tietämystekniikka (knowledge engineering) on tekoälytutkimuksen alue, joka on keskittynyt inhimillisen tiedon esittämiseen, hankkimiseen ja hyväksikäyttöön ongelmien ratkaisuisissa. Asiantuntijajärjestelmät (aj, expert systems) ovat tietokoneohjelmia, jotka pyrkivät ratkaisemaan rajatun alueen ongelmia ihmisasiantuntijoille tyypillisellä tavalla. Tietämystekniikka on tekoälyn kaupallisesti merkittävin ja nopeimmin kasvava sovellusalue. /12/

Tietokoneiden ja erityisesti mikrotietokoneiden nopea kehitys viimeisen vuosikymmenen aikana on mahdollistanut tehokkaiden tietokoneiden hankinnan kohtuullisin kustannuksin. Viime vuosina vallinnut suuntaus siirtyä suurista mainframe-tietokoneista henkilökohtaisiin mikrotietokoneisiin ja työasemiin jatkuu. Mikrotietokoneiden kehitys sekä tietämystekniikan kypsyminen soveltamisen asteelle ovat saaneet aikaan asiantuntijajärjestelmien kaupallisen läpimurron. Markkinoille on ilmestynyt lukuisia mikrotietokoneille tehtyjä asiantuntijajärjestelmien rakentamisvälineitä, aj-kehittämiä, joiden laatu paranee vuosi vuodelta.

Tietekniikassa on lukuisia mahdollisia sovelluskohteita, joissa asiantuntijajärjestelmiä voidaan käyttää. Nykyisissä kehittimissä on mahdollisuus liittää ulkopuolisia atk-ohjelmia asiantuntijajärjestelmiin. Näin saadaan yhdistettyä laskennalliset (kvantitatiiviset) ja laadulliset (kvalitatiiviset) piirteet ongelmista yhdeksi kokonaisuudeksi.

Asiantuntijajärjestelmät on todettu hyväksi välineeksi esittämään rajatun alueen asiantuntemusta 'älykkäällä' tavalla. Perinteisen atk-ohjelman ja asiantuntijajärjestelmän ominaisuuksia on esitetty kuvassa 1. Asiantuntijajärjestelmät soveltuvat suppeiden, korkeatasoista tietoa vaativien ongelmien ratkaisuun, kun perinteiset atk-ohjelmat sovel-

tuvat suurien tietomäärien ja erityisesti numeerisen tiedon käsittelyyn.



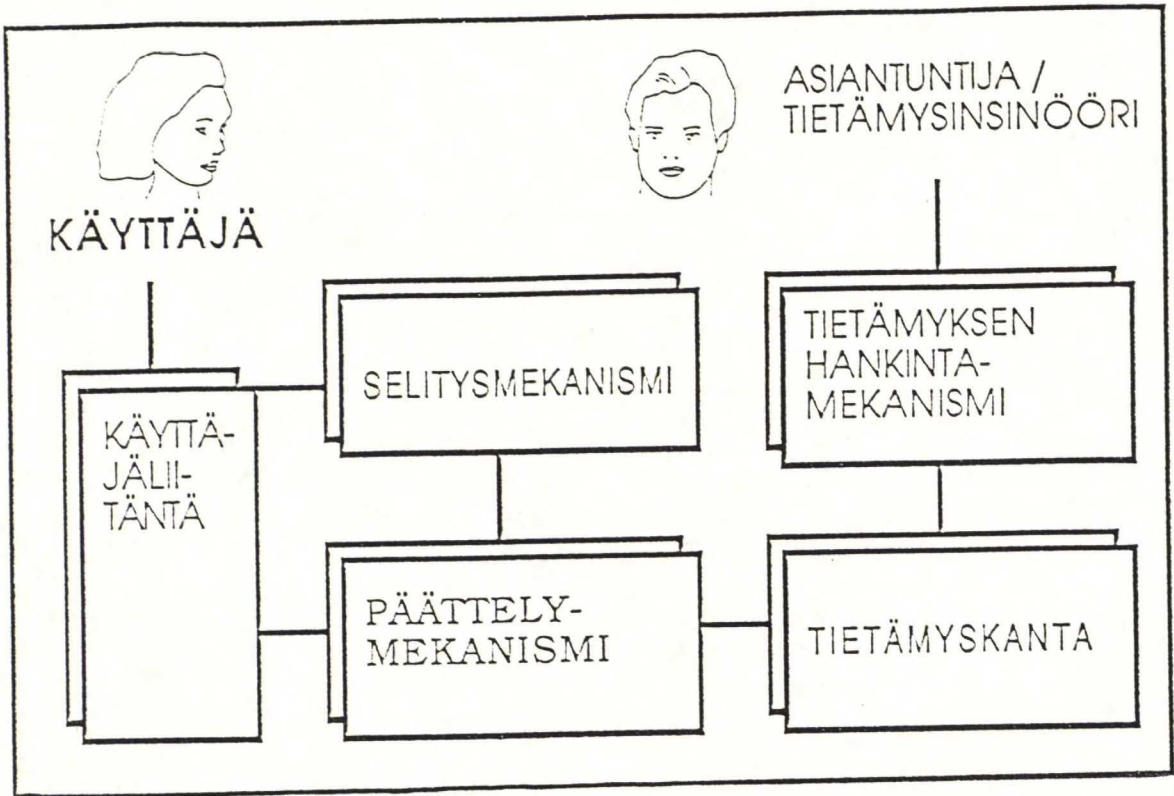
Kuva 1. Asiantuntijajärjestelmien ja perinteisten atk-ohjelmien ominaisuudet /28/

### 1.1 Peruskäsitteet

Asiantuntijajärjestelmä eroaa rakenteeltaan tavallisesta atk-ohjelmasta lähinnä siten, että ohjausmekanismi ja järjestelmän sisältämä tieto sijaitsevat erillään. Asiantuntijajärjestelmän perusrakenne koostuu yleensä seuraavista osista : /19/

- 1) Päättelymekanismi (inference engine).
- 2) Tietämyskanta (knowledge base)
- 3) Tietämyksen hankintamekanismi
- 4) Selitysmekanismi
- 5) Käyttäjäliittyntä





Kuva 2. Asiantuntijajärjestelmän perusrakenne.

Asiantuntijajärjestelmien ohella puhutaan myös tietämysjärjestelmistä (tj, knowledge-based systems). Tj on ohjelmisto tai ohjelma, jonka toiminta perustuu tietämyskannan ja tietämystekniikan menetelmien käyttöön. Asiantuntijajärjestelmän ja tietämysjärjestelmän eroa on vaikea määrittää, mutta yleensä asiantuntijajärjestelmät käsitetään tietämysjärjestelmien osajoukoksi.

## 1.2 Muita käytettäviä termejä /12, 19, 20/

Elinkaari (life-cycle). Ohjelmiston elinkaari alkaa, kun tunnistetaan ohjelmiston tarve ja päättyy, kun ohjelmisto poistetaan käytöstä.

Eteenpäinpäättely (forward chaining) on ohjausstrategia, joka lähtötiedoista edeten pyrkii ongelman ratkaisuun.

Heuristiikka (heuristics) on ongelman ratkaisun metodiikka, joka perustuu kokemusperäiseen tietämykseen. Tällaista tietämystä ovat mm. nyrkkisäännöt (rules of thumb).

Kehitin (tool, shell) on ohjelmisto, jonka avulla asiantuntijajärjestelmiä voidaan rakentaa.

Kehys (frame) on tietorakenne, jota käytetään tietämyksen esittämisessä.

Metatietämys (metaknowledge) on tietämystä itse tietämystä. Metatietämys sisältää ongelmanratkaisuprosessia ohjaavat strategiat ja menetelmät.

Objekti, olio (object) on järjestelmän itsenäinen yksikkö, joka koostuu tiedoista tai toimenpiteistä.

Sääntö (rule) on tietämyksen esittämistapa, joka määrittelee tilanteen ja toiminnon välisen riippuvuussuhteen.

Sääntötulkki (rule interpreter) päättelymekanismin osa, joka tulkitsee ongelmanratkaisuprosessissa käsiteltävät säännöt.

Taaksepäinpäättely (backward chaining) ohjausstrategia, joka lähtee halutusta lopputilanteesta liikkeelle, ja käy sääntöjä takaperin läpi.

Tietämyksen esittäminen (knowledge representation) tekniikka, jolla tietämys mallinnetaan ongelman ratkaisun edellyttämään muotoon.

Tietämysinsinööri (knowledge engineer) on henkilö, joka

suunnittelee ja rakentaa tietämysjärjestelmiä.

Tietämyskanta (knowledge base) on asiantuntijajärjestelmän osa, jossa sovelluksen asiantuntemus sijaitsee. Se on kokoelma tosiasioita, olettamuksia ja sääntöjä.

### 1.3 Tietämystekniikan soveltuvuus tietekniikkaan

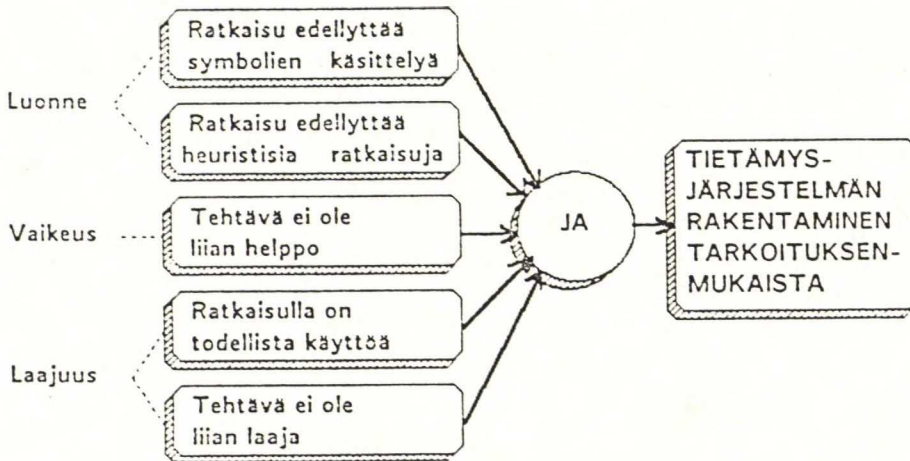
Tietekniikassa on paljon mahdollisia aj-tekniikan sovelluskohteita. Monet tietekniikan käytännön ongelmat vaativat erityistä asiantuntemusta ja kokemuksen mukana tuomia nyrkkisääntöjä, joiden avulla ongelmat voidaan ratkaista. Pieniä mikrotietokonepohjaisia asiantuntijajärjestelmiä voidaan käyttää suunnittelunormien tai asiantuntemusta vaativien pienten suunnittelukokonaisuuksien siirtämiseksi mikroille. Näitä voisi olla vaikka tievalaistuksen valaisimien valinta ja valaisimien sijoitus, bussipysäkkien mitoitus, tielainsäädännön osa-alueet. Näin saataisiin käytössä oleva asiantuntemus useampien työntekijöiden käyttöön ja työskentelyn tehokkuus paranisi. Uusissa kehittimissä on tehokkaat liittynät ulkopuolisiin ohjelmiin, joten järjestelmä voi käyttää jo olemassa olevia laskentaohjelmia erilaisiin laskelmiin.

Asiantuntijajärjestelmän rakentamiseen ei pidä ryhtyä ennen kuin on ensin tutkittu, soveltuuko aj-lähestymistapa ongelman ratkaisuun vai onko jokin muu tapa parempi. Ensimmäinen edellytys on se, että valitun kohdealueen asiantuntijoita on olemassa. Asiantuntijoiden on kyettävä selittämään ratkaisunsa, jotta ne voidaan siirtää tietämyskantaan. Ongelman tulee ratketa tietämykseen perustuvalla päättelyllä. Nyrkkisääntönä voidaan pitää että, jos asiantuntija ei pysty ratkaisemaan ongelmaa muutamassa tunnissa, se on liian monimutkainen. Jos asiantuntijalta menee muutama minuutti ongelman ratkaisemiseen, on valittu ongelma liian

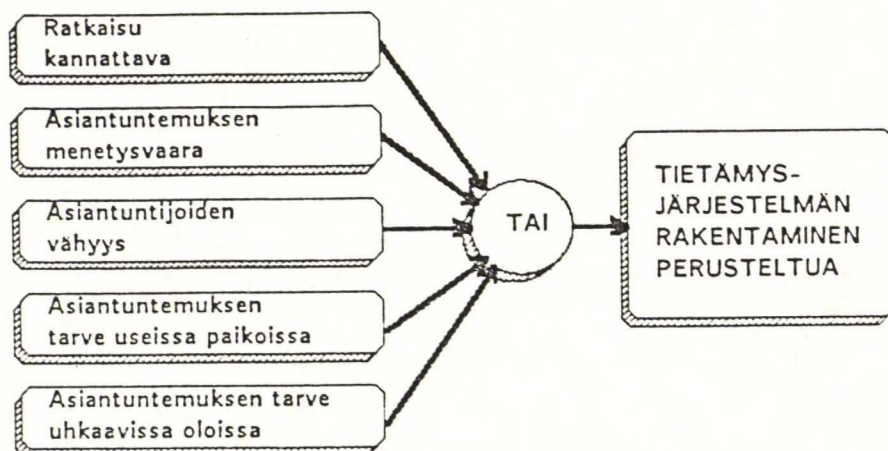


helppo. Aj-lähestymistapa soveltuu symbolien käyttöön perustuvaan päättelyyn ja tilanteisiin, joiden ratkaisussa käytetään heuristiikkaa. Yleensä ongelmat, jotka voidaan ratkaista algoritmisesti tai matemaattisesti ovat tehokkaammin ratkaistavissa muilla apuvälineillä. Tietämystekniikkaa ei tulisi nähdä vaihtoehtona perinteiselle tietotekniikalle vaan sitä täydentävänä uutena tekniikkana. Uudet kehittimet tarjoavat kuitenkin erittäin helpon tavan käsitellä tietämystä ja mahdollisuudet soveltaa aj-tekniikkaa ovat lähes rajattomat./17/

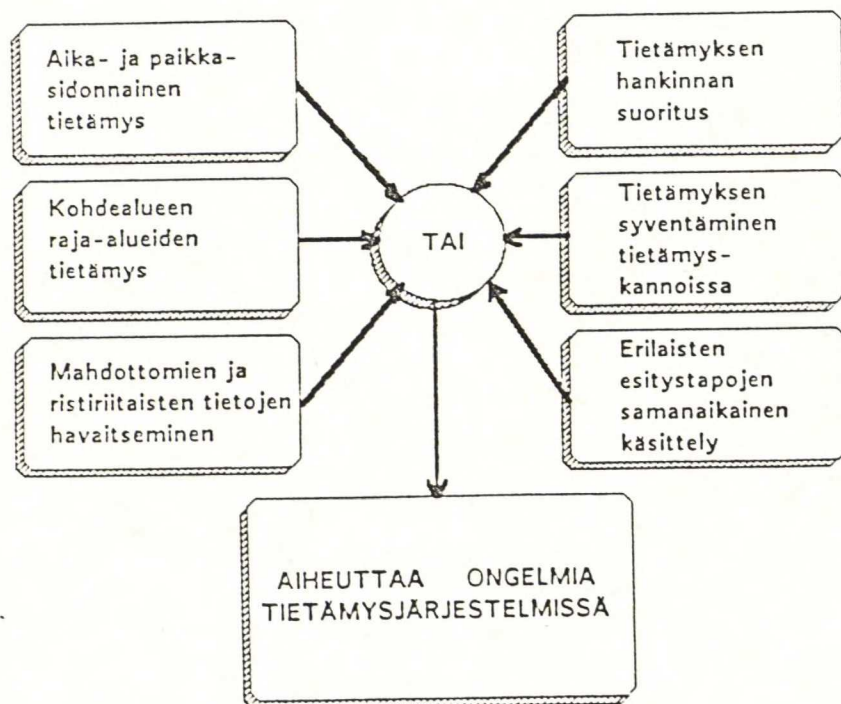
Jos päädytään siihen tulokseen, että asiantuntijajärjestelmän rakentaminen on mahdollista, on sen lisäksi vielä tutkittava onko se tarkoituksenmukaista ja perusteltua (kuva 3). Ongelman, joka ei sovellu aj-tekniikkaan, tyypillisiä piirteitä on esitetty kuvassa 4.



Kuva 3. a) Syyt, milloin asiantuntijajärjestelmän rakentaminen on tarkoituksenmukaista /17/



Kuva 3. b) Syyt, milloin asiantuntijajärjestelmän rakentaminen on perusteltua /17/



Kuva 4. Ongelman, joka ei sovellu asiantuntijajärjestelmä-tekniikkaan, tyypillisiä piirteitä. /17/

## 2. TIETÄMYKSEN ESITTÄMINEN

### 2.1 Yleistä

Tietämyksen esittämistekniikat ovat menetelmiä, joilla sovellusalueen asiantuntemus esitetään tavalla, joka voidaan siirtää asiantuntijajärjestelmän tietämyskantaan. Sovellusalueen tietämys voi esiintyä monessa muodossa. Se voi koostua sovellusalueen kuvauksista, suhteista tai prosedureista. Tietämyskannan kuvaukset tunnistavat oliot ja erottavat ne toisistaan. Kuvaukset koostuvat lauseista, joiden peruskomponentit ovat alkeellisia käsitteitä ja piirteitä. Kuvausjärjestelmässä on yleensä säännöt tai proseduurit, joilla sovelluksien kuvauksia tulkitaan. Tietämyskannassa on myös kuvauksia, joita kutsutaan suhteiksi (relationships). Näillä kuvataan eri olioiden välisiä riippuvaisuuksia. Prosedureilla määritellään eri tehtävien suorittaminen ongelman ratkaisussa. Toisen määrittelyn mukaan tietämystä voidaan kuvata mallilla, jonka komponentteja ovat oliot ja niiden ominaisuudet, tapahtumat, toimintakyky ja metatietämys. Tietämys ei käytännössä ole missään säännöllisessä, määrättyssä muodossa. Asiantuntijalla on päässään eri asioita yhteenliittäviä tosiasioita ja oletuksia : tietämystä, jonka avulla hän pystyy ratkaisemaan oman alansa ongelmia. /10/

Esittämistekniikat voidaan luokitella kahteen pääluokkaan, sen mukaan, edustavatko ne proseduurimaista vai deklaratiiivista esitystekniikkaa (kuva 5).

Deklaratiivinen eli selvittävä esittämistekniikka kuvaa tietämyksen staattisia piirteitä. Näitä ovat tosiasiat kohteesta, tapahtumat ja olioiden väliset relaatiot. Deklaratiivisiin esitystekniikoihin kuuluvat mm. semanttiset verkot, kehystekniikka ja predikaattilogiikka./12, 22/



Proseduurimainen esittämistekniikka kuvaa tietämystä usein ohjelman muodossa. Se kertoo kuinka ongelmaan liittyvää tietoa tulee käsitellä. Produktiotekniikka ja proseduurimainen predikaattilogiikka ovat tunnetuimpia näistä tekniikoista.

| TIETÄMYKSEN ESITTÄMISTEKNIIKAT   |  |
|--|--|
| DEKLARATIIVISET<br>- semanttiset verkot<br>- kehykset<br>- predikaattilogiikka | PROSEDUURIMAISET<br>- produktiotekniikka (säännöt)<br>- proseduurimainen predikaattilogiikka |

Kuva 5. Tietämyksen esittämistekniikat

Tietämyksen esittämistekniikoista logiikka on formaali ja kehykset, semanttiset verkot ja säännöt epäformaaleja tietämyksen esittämistekniikoita. Mikropohjaisissa kehityksissä käytetään yleensä epäformaaleja esittämistekniikoita./22/

## 2.2 Lauselogiikka

Lauselogiikka (propositional logic) koostuu lausekkeista, kuten

|                |    |                    |
|----------------|----|--------------------|
| Pekka on poika | ja | Pekan auto on uusi |
|----------------|----|--------------------|

Jokaiselle lausekkeelle voidaan antaa totuusarvo. Itse ohjelman ei tarvitse tietää mitä lauseet tarkoittavat. Lauselogiikassa voidaan yhdistellä lausekkeita liittimillä,



kuten JA, TAI, negaatio ( NOT ). Lausekkeita voidaan muodostaa loogisten lausekkeiden ja liittimien yhdistelmistä : /12 /

JOS aurinko paistaa JA on lämmintä NIIN ilma on kaunis.

### 2.3 Predikaattilogiikka

Predikaattilogiikka (predicate calculus) on paljon tutkittu symbolirakenteiden kuvaamiskieli, jota voidaan käyttää tietämyksen kuvaamiseen tietokoneissa. Se on laajennus lauselogiikasta, sillä siinä voidaan kuvata yksittäisiä olioita ja niiden välisiä suhteita. Predikaattilogiikassa tietämystä esittävät symbolirakenteet muodostetaan termeistä ja predikaateista. Termeillä kuvataan käsiteltävien olioiden nimiä ja predikaateilla olioiden välisiä relaatioita. / 10, 12/

G 4214 ON karamurskain

ja

karamurskain KUULUU-JOUKKOON murskaimet

lauseissa G4214, karamurskain ja murskaimet ovat termejä kun puolestaan ON ja KUULUU-JOUKKOON ovat predikaatteja.

Predikaattilogiikassa voidaan ilmaista mutkikkaampia kuvauksia funktioiden avulla. Myös liittäjiä ( AND, OR, NOT) tai matemaattisia kvanttoreita (  $\exists$ ,  $\forall$  ) voidaan käyttää. Näillä lisäpiirteillä voidaan kuvata jonkin verran myös epävarmaa tietämystä. /12/

Predikaattilogiikan lausekkeilla voidaan kuvata lähinnä

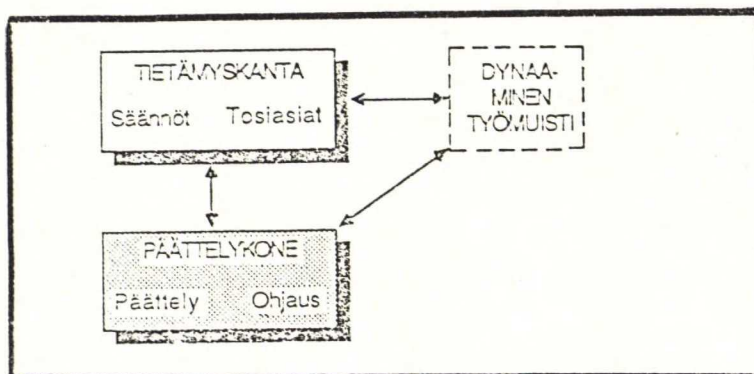
tosiasioita. Uusien rakenteiden muodostaminen onnistuu myös, jos predikaattilogiikan lisäksi käytetään päättelysääntöjä.

### Logiikkapohjaisten järjestelmien ominaisuuksia

Logiikkapohjaisen tietämyksen esittämistekniikan vahvoja puolia ovat luonnollisuus, tarkkuus ja joustavuus sekä modulaarisuus. Heikkoina puolina on pidettävä esitettävien tietorakenteiden yksinkertaisuutta, epävarman tietämyksen puutteellista esittämistä ja päättelyn kulun ohjaamisen heikkoutta. Predikaattilogiikan suosiota lisäsi varmasti osaltaan japanilaisten päätös käyttää Prolog-ohjelmointikieltä viidennen sukupolven tietokoneprojektissaan./6, 12, 25/

### 2.4 Produktiotekniikka

Produktiotekniikka on hyvin yleinen asiantuntijajärjestelmissä käytetty tietämyksen esittämistekniikka. Produktiotekniikka perustuu yhteen yleiseen rakennemalliin, produktiosääntöjen tai JOS-NIIN-sääntöjen hyväksikäyttöön. Perusrakenne koostuu kolmesta peruskomponentista (kuva 6).



Kuva 6. Produktiojärjestelmän rakenne /12/

- 1) dynaaminen työmuisti, joka sisältää ongelman ratkaisun lähtökohdan, kuten tosiasiat, oletukset ja tavoitteet.
- 2) tietämyskanta, joka sisältää järjestelmän tietämyksen ja tosiasiat ongelma-alueesta.
- 3) sääntötulkki, eli järjestelmän päättelymekanismi, joka ohjaa ongelman ratkaisua.

Tietämyskannan tietämys kuvataan säännöillä.

|                        |
|------------------------|
| JOS ehto NIIN toiminto |
|------------------------|

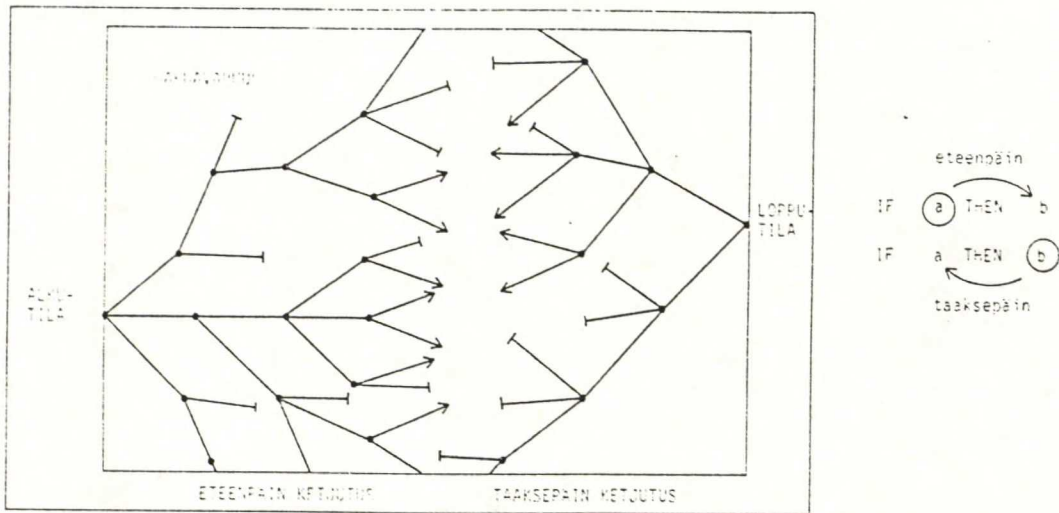
Ehto-osaa kutsutaan Left hand side:ksi (lhs) ja toiminto-osaa right hand side:ksi (rhs). Ehto-osa testaa nykytilan ominaisuuksia, ja jos ne toteutuvat sanotaan, että sääntö laukeaa, eli oikean puolen toiminto-osa toteutetaan. Toiminto-osa voi sisältää alkeistoiminnon tai johtopäätöksen ja se muuttaa nykytilaa./12/

Sääntötulkki on tietämysjärjestelmän päättelykoneen tärkein komponentti. Se ratkaisee mitä sääntöjä kulloinkin otetaan käyttöön. Sääntötulkin ohjausstrategia voi olla eteenpäin- tai taaksepäinpäättelevä. Useimmat mikropohjaiset kehittimet ovat taaksepäin päätteleviä, mutta niissä on usein mahdollisuus myös kaksisuuntaiseen päättelyyn.

Eteenpäinpäättelyssä (tieto-ohjattu) sääntötulkki käy läpi kaikki säännöt annettujen lähtötietojen perusteella. Ne säännöt, joiden ehto-osa toteutuu aktivoidaan. Sääntöjen toiminto-osien suorittamisesta saadut uudet tiedot talletetaan työmuistiin seuraavaa laukaisukierrosta varten. Toimintaa jatketaan, kunnes kaikki tietämys on käyty läpi tai tavoite on saavutettu (kuva 7).



Taaksepäinpäätelyssä (tavoiteohjattu) ratkaisun lähtökohta on jokin tavoite. Sääntötulkki aktivoi ne säännöt, joiden toiminto-osan johtopäätöksenä on oletettu tavoite. Sääntöjen laukeamisen edellytyksenä on ehto-osan toteutuminen. Päätelyketjua edetään, kunnes vaadittu tavoite on saavutetaan tai todetaan se mahdottomaksi (kuva 7).



Kuva 7. Eteenpäin- ja taaksepäinpäätely /6/

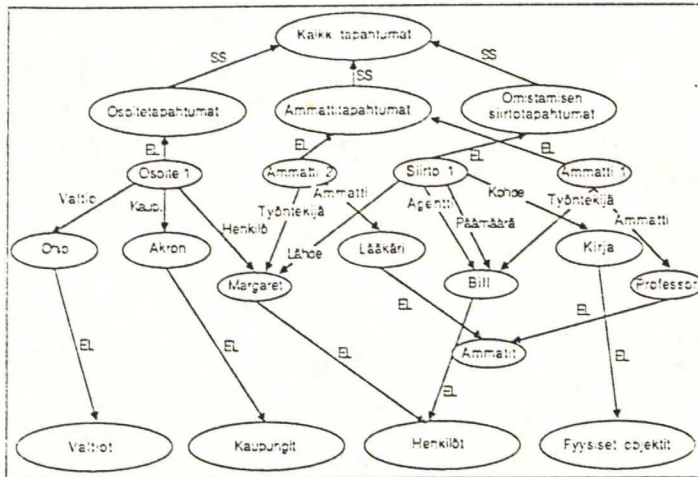
#### 2.4.1 Produktiotekniikan ominaisuuksia

Produktiotekniikan hyviä puolia ovat tietämyksen modulaarisuuden ansiosta helppo muunneltavuus, selkeys ja luonnollisuus. Tietämyskantaan on helppo lisätä uutta tietämystä. Produktiotekniikassa sääntöjen rakenne on helppotajuinen. Haittapuolia ovat ohjelman toiminnan hitaus sekä päätelyn kulun seurannan vaikeus./12/

#### 2.5 Semanttiset verkot

Semanttiset verkot ovat yhteisnimitys tietämyksen esittämistekniikoille, jotka esittävät tietämystä verkkorakenteiden

avulla. Semanttisessa verkossa oliot, tilanteet esitetään solmuina ja solmujen väliset suhteet (relaatiot) nuolilla (kuva 8).

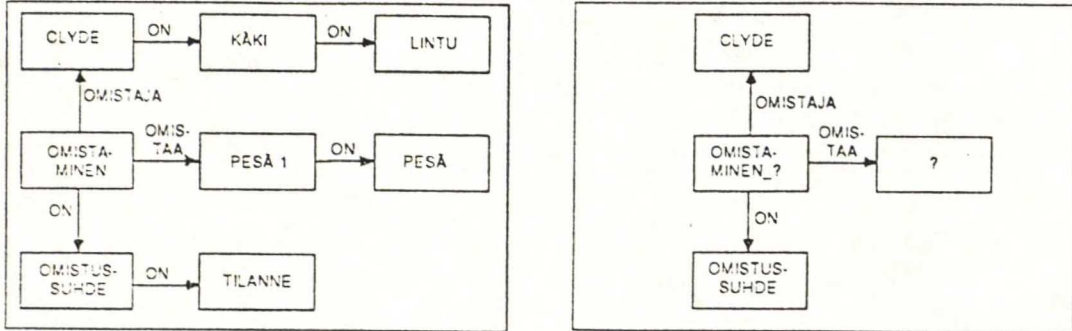


Kuva 8. Semanttinen verkko.

Semanttisten verkkojen käyttö mahdollistaa uusien tosiasioiden johtamisen annetun hierarkisen esitysrakenteen pohjalta. Tätä kutsutaan periytymiseksi. Tästä syystä semanttisia verkkoja käytetään mutkikkaita taksonomiarakenteita vaativissa sovellutuksissa. /12/

## 2.51 Semanttisten verkkojen toimintaperiaate

Semanttisessa verkkoesityksessä ei ole ennalta sovittu, mitä kuvattu rakenne tarkoittaa. Verkon rakenteelle määritellään merkitys niillä proseduureilla, jotka hyväksikäyttävät verkkoa. Useimmiten järjestelmät perustuvat verkkorakenteiden sovittamiseen (matching). Sovituksessa muodostetaan kyselyä varten oma osaverkko, joka sovitetaan tietämyskannan verkkoa vasten, jolloin se etsii löytyykö sieltä haluttu kohde (kuva 9).



Kuva 9. Sovitus eli matching.

## 2.52 Semanttisten verkkojen ominaisuuksia

Semanttiset verkot ovat yleinen tiedon esittämistapa. Ne soveltuvat hierarkisen tiedon ja laajojen luokitusverkkoja sisältävien sovelluksien esittämiseen. Niiden huonoina puolina on pidettävä puutteita epävarman tietämyksen kuvaamisessa sekä semantiikan ongelmia, kuten mitä solmu todella tarkoittaa.

## 2.6 Kehystekniikka

Kehystekniikka on uusimpia tietämyksen esittämistapoja. Se muodostuu kehyksistä (frames), joihin kerätään kaikki tiettyyn kohteeseen liittyvä tietämys. Yleisimmässä kehystekniikassa muodostetaan kutakin oliota varten kehys, jossa on tarvittavat lokerot (slots) olioille liittyville tiedoille. Kuvassa 10. selvitetään esimerkin avulla kehystekniikan käyttöä.



|      |          |                |
|------|----------|----------------|
| Auto | merkki   | SAAB           |
|      | omistaja | VILLE VIRTANEN |
|      | väri     | PUNAINEN       |
|      | renkaat  | 4 kpl          |

Kuva 10. Kehys lokeroineen.

Kehystä voidaan kuvata myös useamman tasoisena verkkona, jossa ylätasot kuvaavat kehyksen yleisiä ominaisuuksia ja alatasot eli lokerot kuvaavat tapauskohtaista erityistietämystä. Yleensä lokeroille on annettu oletusarvot, joten kehystä voidaan käyttää, vaikka joka lokeron sisältöä ei olisikaan määriteltä. Tämä on tärkeä ominaisuus, sillä ihmisetkin tulkitsevat uusia asioita sen pohjalta, millaisia kokemuksia heillä on entuudestaan. Esimerkiksi jos ostetaan auto, oletetaan, että siinä on neljä pyörää, moottori ja istuimet matkustajille. /12, 21/

#### 2.61 Proseduraalinen tietämys kehystekniikassa

Kehystekniikka luokitellaan staattiseksi tietämyksen esittämistavaksi, mutta siinä useimmiten on mahdollisuus proseduraalisen tietämyksen esittämiseen. Tämä on toteutettu liittämällä lokeroihin proseduureja, jotka voivat ohjata järjestelmän ongelmanratkaisumekanismia. /12/

#### 2.62 Kehystekniikan toimintaperiaate

Kehysjärjestelmässä tiettyä tilannetta edustavan kehyksen lokerot täytetään tilannetta kuvaavilla tiedoilla tai niiden puuttuessa oletusarvoilla. Lokerot voidaan täyttää myös



kyselemällä käyttäjältä puuttuvia tietoja, tai lokerot voivat saada arvon myös periytymällä. Jos lokerossa on proseduuri käynnistetään se. Järjestelmä voi myös tutkia alikehysten sisällön. /12/

### 2.63 Kehystekniikan ominaisuuksia

Kehystekniikka soveltuu mutkikkaiden rakenteiden ja hierarkioiden kuvaamiseen. Tiettyyn olioon liittyvä tietämys on koottuna yhteen kehykseen ja on helposti näkyvillä. Tämä helpottaa tietämyksen käsittelyä. Kehystekniikan heikkoihin puoliin kuuluu puutteellinen proseduraalisen tiedon käsittely. Kehystekniikan puutteita on pyritty poistamaan liittämällä järjestelmään produktiotekniikan käyttömahdollisuus. /12/

### 2.7 Muita tietämyksen esittämistekniikoita

Analoginen tietämyksen esittämistekniikka käsittää analogiset menetelmät esittää suoraan tietämys mahdollisimman luonnollisessa muodossa. Se soveltuu hyvin graafisen tietämyksen, kuten karttojen, diagrammien ja piirrosten esittämiseen. Sen etuna ovat esityksen havainnollisuus. Heikkona puolena sillä on puutteellinen epävarman tietämyksen esittäminen. /12/

Toiminnallisessa esittämistekniikassa erotetaan tietämyksen esittämisessä kaksi seikkaa: mitä tietämyskanta tekee järjestelmälle ja kuinka tietämyskanta esittää sen, mitä se tietää. Toiminnallisessa tietämyksen esittämistekniikassa on yhdistetty kehystekniikan ominaisuudet rakenteen kuvaamisessa ja predikaattilogiikan ominaisuudet päättelymekanismin toteutattamisessa. Tämä on melko uusi ja vähän käytetty tekniikka. /12/

## 2.8 Esittämistekniikoiden yhdistely

Mikään edellä mainituista tekniikoista sopii harvoin kaikilta ominaisuuksiltaan sovellusalueen tietämyksen esittämistehtäviin. Uusissa mikropohjaisissa hybridikehittimissä onkin yleensä useita tietämyksen esittämistekniikoita yhdistetty. Tulevaisuudessa on odotettavissa kehittämiä, joissa on useita tietämyksen esittämistekniikoita, jotka sopivat toistensa kanssa hyvin yhteen./19/

## 2.9 Tietämyksen esittämisen tasot

Tietämyskanta voidaan järjestää erottamalla tietämys eri tasoille. Yleensä tietämys jaetaan kolmelle tasolle :

- 1) Tosiasiat
- 2) Heuristinen tietämys
- 3) Metatietämys

Heuristinen tietämys on kokemusperäistä tietoa, jota kuvataan em. produktiosäännöillä.

Metatietämys on tietämystä itse tietämyksestä. Metatietämyksen avulla ohjataan ongelmanratkaisuprosessia. Se sisältää ne menetelmät ja strategiat, joita prosessi käyttää :

- \* päättely- tai ohjausstrategiavaihtoehdot (eteen- tai taaksepäin)
- \* kuinka relevantti tietämys aktivoidaan
- \* kuinka useammasta vaihtoehdosta valitaan sopivin
- \* kuinka uutta tarvittavaa tietämystä hankitaan
- \* kuinka varmuuskertoimet määritellään

### 3. ASIAANTUNTIJAJÄRJESTELMÄN RAKENTAMINEN

Perinteisesti asiantuntijajärjestelmän rakentaminen on ollut asiantuntijan ja tietämysinsinöörin yhteistyötä, jossa asiantuntija kertoi tietämysinsinöörille tietonsa ja tietämysinsinööri rakensi asiantuntijajärjestelmää kuvaamalla asiantuntijan tietämystä järjestelmän tietämyskantaan. Nykyaikaisilla mikropohjaisilla kehittimillä pystyvät asiantuntijat yksinäänkin rakentamaan omia järjestelmiään. Laajojen asiantuntijajärjestelmien rakentaminen vaatii kuitenkin perehtymistä tietämystekniikan keinoihin./7/

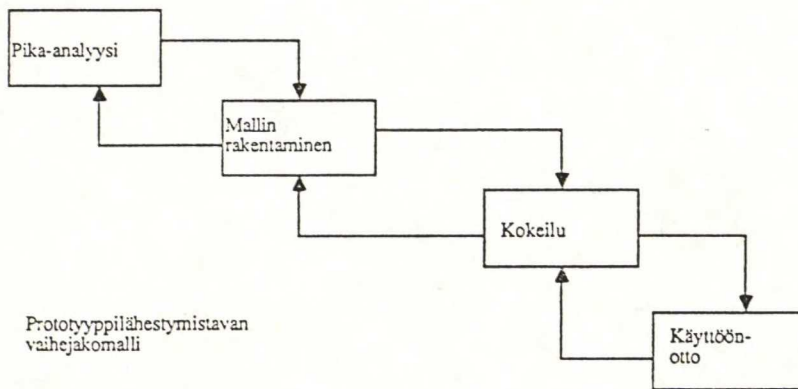
Asiantuntijajärjestelmien rakentaminen ja tietämyksen hankinta kuuluvat erottamattomasti yhteen. Prosessit ovat päällekkäisiä toistensa kanssa, ja käytännössä tapahtuvat samaan aikaan. Selvyyden vuoksi ne käsitellään tässä eri luvuissa, vaikka jotkin asiat tulevatkin toistetuksi.

#### 3.1 Yleistä

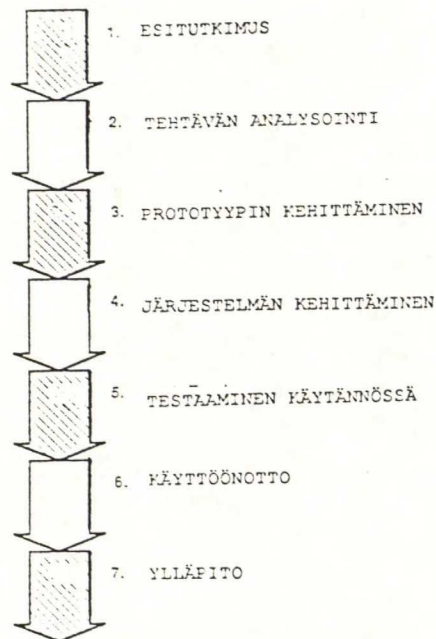
Asiantuntijajärjestelmän rakentaminen sisältää useita vaiheita, joista monet muistuttavat perinteisen ohjelmistoprojektin vaiheita. Aj-kehitysprojekteissa käytetään usein prototyyppilähestymistapaa (kuva 11) koska pieniä prototyypppejä voidaan rakentaa nopeasti ja järjestelmää voidaan testata keskeneräisenäkin. Tämä on mahdollista, koska ongelma-alueen tietämys ja päättelyä ohjaava mekanismi sijaitsevat erillään. Perinteisissä atk-ohjelmissa tiedot ja ohjelman ohjaus (kontrolli) ovat molemmat samassa ohjelmakoodissa. Kun prototyypppejä tehdään tiheästi ja testataan asiantuntijalla tai käyttäjillä he voivat ilmoittaa välittömästi, jos tietämysinsinööri on ymmärtänyt jonkun kohdan väärin. Se osa voidaan tehdä uudestaan, ilman että suurta työpanosta on mennyt hukkaan. Tällaisen kehitysprosessin aikana asiat selkiytyvät sekä tietämysinsinöörille että käyttäjille ja järjestelmää voidaan tarkentaa prosessin



edetessä. Pienten ja suurten aj:den kehittämisprojekteja voidaan kuvata samanlaisella peruskaaviolla, joskin on selvää, että jotkut asiat painottuvat eri kokoisilla projekteilla erilailla. Tässä käsitellään sekä pienten että suurempien mikropohjaisten järjestelmien rakentamisvaiheet, sillä usein pienestä aj:stä kehitetään ajan kuluessa laajempi versio. Mikrotietokoneiden kapasiteetin parantuksessa, tulevaisuudessa rakennetaan mikroillakin suuria aj:ä. Kuvassa (12) on aj-kehitysprojekti jaettu 7 eri vaiheeseen./7, 20/



kuva 11. Prototyyppilähestymistavan vaihejako /20/



kuva 12. Asiantuntijajärjestelmän rakentamisvaiheet /7/

### 3.2 Pienten mikropohjaisten asiantuntijajärjestelmien rakentamisvaiheet

Pienet mikropohjaiset järjestelmät ovat usein käyttäjän henkilökohtaisia apuvälineitä. Niillä pyritään nopeasti saamaan toimiva järjestelmä, jota voidaan ajan myötä täydentää. Oman aj:n rakentaminen on myös paras tapa opetella tietämystekniikkaa. Seuraavassa jaottelussa luetellaan lyhyesti pienen aj:n rakentamisvaiheet. Suurempien mikropohjaisten aj:ien kehitysprojektit noudattelevat kohdassa 3.3 selostettua vaihejakoa./20/

1. Tunnistetaan sopiva ongelma ja analysoidaan sen ratkaisulle asetettavat vaatimukset.
2. Valitaan aj:n rakentamisen työkalu ja sitoudutaan tiettyyn konsultointityyppiin: diagnostisointi, valinta tms.
3. Kerätään ja analysoidaan ongelmaan liittyvä tietämys.
4. Suunnitellaan asiantuntijajärjestelmä : luonnostellaan aj:n rakenne paperille lohkokaavioina yms. kuvauksina ja muotoillaan muutamia keskeisiä päättelysääntöjä.
5. Toteutaan (demonstraatio)prototyyppi valitulla apuvälineellä: luodaan tietämyskanta ja testataan se muutamien testitehtävien avulla.
6. Laajennetaan, testataan ja korjataan prototyyppiä kunnes se suoriutuu ongelmatehtävästä halutulla tavalla
7. Ylläpidetään ja päivitetään prototyyppiä tarpeen mukaan.

### 3.3 Keskikokoisten asiantuntijajärjestelmien rakentamisvaiheet

#### 3.31 Esitutkimus

Esitutkimuksen päätarkoitus on valita sopiva sovellusongelma ja tutkia kaikki seikat, joilla voi olla vaikutusta projektiin, ennen kuin se käynnistetään. Näitä voivat olla

- 1) onko järkevää tehdä aj
- 2) onko taloudellisesti kannattavaa tehdä aj
- 3) millaista laitteistoa tarvitaan
- 4) millaista ohjelmistoa tarvitaan
- 5) millaisia työntekijöitä tarvitaan

Tässä vaiheessa on myös tarkasti määriteltävä mitä aj:ltä halutaan eli koko projektin päämäärä ja tavoitteet. Tavoitteita voidaan pitää opasteina koko projektin läpi. Myös asiantuntijoiden ja käyttäjien asettamat vaatimukset on määriteltävä. Jos kysymyksessä ei ole aivan pieni projekti, on syytä kiinnittää huomiota projektin johtoon. Projektin vetäjällä tulee olla kokemusta vastaavan projektin johdosta. Hänen ei tarvitse olla perillä jokaisesta teknisestä yksityiskohdasta, mutta projektin onnistumisen kannalta on välttämätöntä, että hän kykenee johtamaan projektin kulkua. Projektin osallistujat ja muut resurssit määritellään. Tässä vaiheessa on hyvä tehdä karkeahko kustannus-hyöty-analyysi./7/

#### 3.32 Tehtävän analysointi

Projektin toisessa vaiheessa selvitetään, kuinka ongelma on aikaisemmin ratkaistu, neuvotellaan asiantuntijoiden kanssa ja asetetaan lopulliset tavoitteet ja vaatimukset järjestelmälle. Suurille ja keskikokoisille aj-projekteille on



tärkeää määritellä tehtävä, jota varten aj tehdään, erittäin hyvin. Asiantuntijoiden ja käyttäjien kanssa käytävissä keskusteluissa käydään läpi ongelman ratkaisun päävaiheet ja pyritään rajaamaan tehtävä. Käyttäjiltä kysytään millaista tietämystä he tarvitsevat ongelman ratkaisemisessa. Tutkitaan kuinka aj parhaiten sopii olemassaolevaan työympäristöön ja selvitetään myös tarvittava tietämyksen taso. Näissä keskusteluissa on myös etsittävä tietolähteet eli ketkä toimivat pää-asiantuntijoina ja mitä kirjallisuutta, normeja yms. tarvitaan. Tässä vaiheessa voidaan tehdä alustavia laitteisto- ja työkaluvalintoja, tutkimalla minkä tyyppiset aj-kehittimet tehtävään sopivat ja onko kenelläkään kokemusta tietystä kehittämisestä. On korostettava vielä, että ongelmaa ei saa muokata jollekin kehittämiselle sopivaksi, vaan on löydettävä ongelmaan sopiva kehitin./18, 20/

### 3.33 Prototyypin kehittäminen

Projektin kolmannessa vaiheessa aj:stä tehdään prototyyppi, jolla tutkitaan järjestelmän toteutettavuus. Sen rakentaminen on iteratiivinen prosessi. Prototyyppejä tehdään useita ja ne testataan käyttäjillä. Näin saadaan nopea palaute, toimiiko se oikein ja mihin suuntaan järjestelmää tulisi kehittää. Tätä kutsutaan nimellä 'rapid prototyping'. Vaikka se onkin osoittautunut tehokkaaksi, kuvastaa se osaltaan aj:en rakentamismenetelmien kehittymättömyyttä. /7, 20/

Tässä vaiheessa tehdään lopulliset päätökset laitteistosta, kehittämisestä sekä tietämyksen esittämistavasta (esim. kehykset tai säännöt). Tiedonkeräämisstrategia määritellään myös lopullisesti ja se sovitetaan yhteen laitteiston ja ohjelmiston vaatimusten mukaisesti. Tiedon syöttö voi olla tapahtua manuaalisesti tai elektronisesti esim. ulkopuolisen ohjelman tulostuksena. Myös järjestelmän mahdolliset liitännät ulkopuolisiin ohjelmiin (tietokanta, Fortran-,



Pascal-ohjelma) on hahmoteltava ja selvitettävä millaiset ominaisuudet kehittämissä on näitä varten.

### 3.34 Järjestelmän kehittäminen

Suurin osa lopullisen järjestelmän tietämyksestä siirretään järjestelmään tässä vaiheessa. Asiantuntijoiden kanssa käydyistä keskusteluista ja haastatteluista saatu tietämys lisätään tietämyskantaan. Käyttäjäliityntää parannellaan ja verrataan sitä sekä muita järjestelmän ominaisuuksia niihin vaatimuksiin, jotka määriteltiin projektin ensimmäisessä vaiheessa. Asiantuntijan tietämystä kuvaavat säännöt jalostetaan yhdistelemällä ja uudelleenjärjestämällä tietämyskannan sisältö./7/

### 3.35 Testaaminen käytännössä (Field testing)

Järjestelmä täytyy testata todellisessa käyttöympäristössään ja verrata aikaisemmin asetettuja vaatimuksia nykyisiin ominaisuuksiin. Jos puutteita havaitaan, tulee ne korjata.

Aj testataan useita kertoja usealla eri tavalla ja monilla erilaisilla tapauksilla. Testaamisen tulee kohdistua käyttäjäliityntään ja ongelma-alueisiin. Päättelytekniikoita ja tietorakenteita muokataan niin, että ne esittävät tietämystä parhaalla mahdollisella tavalla. Jos varmuusker-toimia käytetään on niiden arvot ja käyttötapa tarkistettava./7/

Ensimmäiseksi tarkistetaan toimiiko järjestelmä virheettömästi. Prototyypin antamia vastauksia verrataan oikean asiantuntijan vastauksiin suorituksen arvostelemiseksi. Testattavat tapaukset tulisi valita niin, että ne ovat tuttuja 'perustapauksia', joista saadaan nopeasti selvitettyä tietämyskannan mahdolliset puutteet. Ihanneolosuhteissa aj:n

vastaukset tulisi olla 80-90 %:sesti samanlaisia kuin asiantuntijan.

Tässä vaiheessa tehdään mahdolliset suuret muutokset järjestelmään. Jos muutoksia tehdään, palataan takaisin prototyypin kehittämisvaiheeseen ja käydään kaikki vaiheet uudelleen läpi. Tämän iteraatioprosessin merkitystä ei voida yliarvioida, varsinkin jos kysymyksessä on suuri aj. Pienikin muutos voi saada aikaan ketjureaktion, jonka vaikutusta ei heti huomata. Sen vuoksi on erittäin tärkeää kerrata nämä vaiheet./7/

### 3.36 Käyttöönotto

Seuraava vaihe on järjestelmän käyttöönotto todellisessa käyttäjän työympäristössä. Tähän vaiheeseen kuuluu käyttäjien koulutus ja siihen voi kuulua myös järjestelmän siirtäminen lopullisen laitteistoon, missä sitä tullaan käyttämään. Käyttäjien koulutus on erittäin tärkeä vaihe, sillä siitä riippuu paljon hyväksyvätkö he sen vai eivät, toisin sanoen onnistuuko projekti vai ei. Tämän vaiheen vaikeudet ovat enemmän psykologisia ja organisaatiollisia kuin teknisiä. Tässä vaiheessa huomataan kaikkien kokouksien, vaiheiden uudelleen suorittamisen ja demonstraatioiden hyöty./7/

### 3.37 Ylläpito

Ylläpitoon tulee tulevaisuutta ajatellen kiinnittää tarpeeksi huomiota, sillä toimivan järjestelmän tietämyskantaa tulee täydentää ja päivittää aina tarvittaessa. Tiedonhankinta prosessi jatkuu koko järjestelmän elinkaaren läpi, seuraavista syistä :/7, 19/

Asiantuntijat eivät muista yhden ongelman kaikkia tärkeitä seikkoja, ennenkuin he joutuvat juuri sen ongelman osan

eteen, joka auttaa heitä muistamaan ao. tapauksen.

Asiantuntijat eivät tiedä kaikkea, vaan he hankkivat koko ajan itse uutta tietämystä.

Tietämysinsinöörit eivät aina pysty hankkimaan kaikkea relevanttia tietämystä asiantuntijoilta. Tämän vuoksi osa asiantuntijan tietämyksestä saattaa puuttua järjestelmän tietämyskannasta.

Tehtävät ja ehdot muuttuvat ajan myötä. Uusi tietämys ongelmasta on sisällytettävä tietämyskantaan, jos järjestelmä halutaan pitää ajan tasalla.

Odottamattomat tilanteet vaativat lisätietämyksen sisällyttämistä tietämyskantaan.

Käyttäjien opittua käyttämään järjestelmää ja ymmärtämään asiantuntijajärjestelmiä, he vaativat enemmän.

### 3.4 Tietämyksen hankinta

Tietämyksen hankinta (knowledge acquisition) on tärkein vaihe asiantuntijajärjestelmän rakentamisessa. Tietämyksen hankintaan osallistuu yleensä sekä asiantuntija että tietämysinsinööri, joiden yhteistyönä järjestelmän tietämyskanta syntyy. Perusajatuksena kirjallisuudessa esitetään usein se, että tietämysinsinööri mallintaa asiantuntijan tietämyksen järjestelmän tietämyskantaan. Tosiasiassa tietämysinsinööri mallintaa oman tietämyksensä, eli kuinka hän on itse ymmärtänyt ongelman ja sen ratkaisemisen. Tässä piilee suuri riski, sillä tietämysinsinöörin harteille sysätään suuri vastuu, jos oletetaan hänen pystyvän omaksumaan asiantuntemus muutamalla tapaamisella asiantuntijan kanssa, kun todellisella asiantuntijalla siihen on mennyt vuosia. Tietämyskannan



rakentamisessa tulisi pyrkiä mahdollisimman läheiseen yhteistyöhön tietämysinsinöörin ja asiantuntijan välillä. Tietämyksen hankinta voi tapahtua myös kirjallisuudesta, kuten normeista, taulukoista tms./ 2, 19, 22/

Tietämyksen hankinnan eri menetelmiä :

Itsetutkiskelu eli introspektio (introspection) tarkoittaa sitä, että asiantuntija tutkii itse omaa päätöksentekoaan ja myös rakentaa tietämyskannan. Ajatuksena se, että henkilö, jolla on ongelman ratkaisun tieto-taito hallussaan on paras henkilö rakentamaan tietämyskanta on hyvä, mutta käytännössä tästä aiheutuu usein seuraavanlaisia ongelmia. /2/

Asiantuntijat eivät aina pysty selittämään omaa tietotaitoaan. Hyväksi havaittu keino on ottaa mukaan joku järjestelmän tulevista käyttäjistä, jolla ei ole ongelmaan liittyvää asiantuntemusta, jolle asiantuntija joutuu selvittämään kuinka ongelman ratkaisu etenee. Tällöin asiantuntija joutuu itse tutkiskelemaan omaa ajatteluaan ja päätöksenteon eri vaiheet selvenevät hänellekin.

Asiantuntijat olettavat usein, että käyttäjät ymmärtävät ongelman paremmin kuin he todellisuudessa ymmärtävät. Usein asiantuntijan rakentama aj soveltuu ainoastaan muiden asiantuntijoiden käytettäväksi, koska sen käsitteet ja ajatusmaailma liikkuu korkeammalla tasolla kuin tavalliselta työntekijältä voidaan odottaa.

Asiantuntijoilla ei ole riittävästi aikaa käytettävänä aj:n kehittämiseen. Työskentely kokeneen tietämysinsinöörin kanssa vähentää aj:n rakentamiseen kuluvaa aikaa huomattavasti.

Lopputuloksena syntynyt järjestelmä saattaa olla kömpelö ja kehittymätön, koska asiantuntijalla ei ole ollut kokemusta

aj:n rakentamisesta.

Haastattelu (interviewing) on paljon käytetty tietämyksen hankintatekniikka. Jokaisessa aj-kehitysprojektissa käytetään haastattelua jonkin verran. Usein tietämyksen hankinta haastattelemalla tapahtuu siten, että tietämysinsinööri keskustelee asiantuntijan kanssa ja kyselee asiantuntijalta tehtävään liittyviä kysymyksiä. Tietämysinsinöörit kirjoittavat muistiinpanoja tai äänittävät haastattelun nauhalle myöhempää analysointia varten. Todellinen tietämyksen irrottaminen asiantuntijan puheesta ja sen muokkaaminen järkevään muotoon jää siten tietämysinsinöörin tehtäväksi. Käytännössä tämä tarkoittaa sitä, että tietämyskanta rakennetaan siltä pohjalta, mitä tietämysinsinööri on tajunnut ongelmasta. Järjestelmät, joiden tietämys on hankittu pelkästään haastattelutekniikkaa käyttämällä, toimivat usein kuten todellinen asiantuntija, mutta eivät hyödynnä asiantuntijan todellista asiantuntemusta. Haastattelutekniikka on ollut ainoa käytettävissä oleva tekniikka silloin, kun aj-rakentamisvälineet ovat olleet niin kehittymättömiä, etteivät niitä osanneet käyttää muut kuin atk-ammattilaiset./2/

Yksi tapa saada selville, kuinka asiantuntija työskentelee on tarkkailla häntä työssään. Asiantuntijan työskentely voidaan kuvata videonauhalle ja pyytää asiantuntijaa itse kommentoimaan sitä jälkikäteen. Kommentointi on tehtävä mahdollisimman pian kuvauksen jälkeen, etteivät asiat pääse unohtumaan. Tarkkailulla ei yksinään ole paljon merkitystä, sillä suurin osa informaatiosta saadaan asiantuntijan puheesta. Tarkkailutekniikan etu on siinä, ettei tietämysinsinöörin tarvitse keskeyttää asiantuntijan työskentelyä. Todellisen asiantuntemuksen tunnistaminen on vaikeaa. Kuvattaessa asiantuntijat saattavat tuntea olonsa kiusaantuneeksi ja toimivat tavalla, joka ei ole luonnollista heille. Tarkkailu-tekniikkaa on käytetty mm.



matalalla lentävän helikopterin ohjaamisen tutkimisessa. /2/

Prototyyppilähestymistapa (prototyping) on haastattelutekniikasta kehitetty tietämyksen hankintatekniikka. Prototyyppilähestymistekniikka eroaa siinä, että asiantuntijajärjestelmän prototyyppi on väline, jonka avulla keskustellaan. Ensimmäiset prototyypit pyritään rakentamaan mahdollisimman nopeasti ja niitä käyvät läpi tietämysinsinööri ja asiantuntija yhdessä. Prototyypppeihin lisätään uutta tietämystä pala palalta. Asiantuntija kokeilee järjestelmän prototyyppiä erilaisilla koeajoilla ja kertoo mitä puutteita tai virheitä siinä on. Tämän tekniikan ydin on siinä, että asiantuntija joutuu perustelemaan ja selittämään ratkaisujaan erittäin yksityiskohtaisissa tapauksissa. /2/

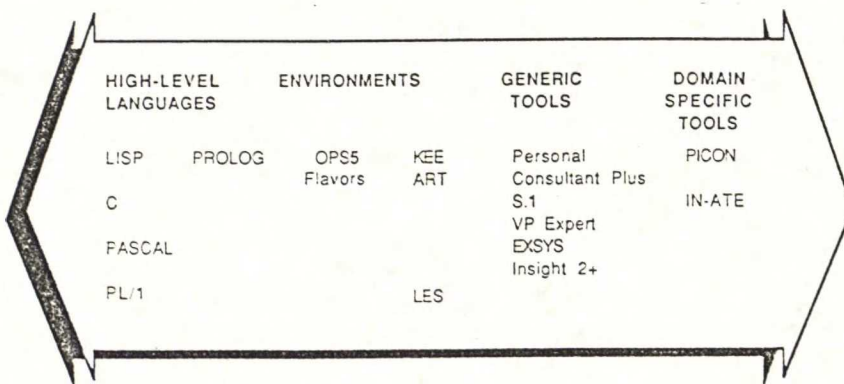
Induktio (induction) on tietämyksen hankintamenetelmä, jolla yritetään vanhoista tapauksista ennustaa tulevia tapahtumia. Induktio on erittäin tehokas menetelmä, jos sopivia esimerkkitapauksia on runsaasti valmiina. Induktio-algoritmi tekee esimerkkitapauksista automaattisesti säännöt tai päättelypuun. Induktiossa piilevät kuitenkin omat vaaransa, sillä ongelmasta luotujen sääntöjen ja ongelman ymmärtämisen välillä ei välttämättä ole mitään yhteyttä. /2/



#### 4. ASIAANTUNTIJAJÄRJESTELMIEN RAKENTAMISVÄLINEET

##### 4.1 Yleistä

Asiantuntijajärjestelmien rakentamisvälineitä ovat ohjelmointikielet, kehitysympäristöt ja kehittimet. Perinteisillä ohjelmointikielillä, kuten C:llä ja Pascalilla voidaan myös rakentaa aj:ä. Helpommin se käy uudenaikaisemmilla ohjelmointikielillä, kuten Prologilla tai LISP:llä. Näissä kielissä on paremmat mahdollisuudet esittää tietämystä. Kehittyneempi rakentamisväline on aj-kehitysympäristö (environment), tähän ryhmään kuuluu mm. OPS5, jolla on suhteellisen helppoa rakentaa aj. Kehitysympäristö on tavallaan ohjelmointikielen ja kehittimen välimuoto. Se antaa mahdollisuudet tehdä juuri halutunlainen asiantuntijajärjestelmä, mutta sen ominaisuudet helpottavat huomattavasti tietämyksen ja tietorakenteiden esittämistä. Helpoin ja nopein tapa rakentaa aj on käyttää aj-kehittämiä, joita tässä luvussa lähinnä tarkastellaan. Kehittimet, joita kutsutaan myös työkaluiksi (tools) sisältävät valmiin päättelymekanismin, selitysmekanismin ja kehittäjäliittynän, mutta tietämyskantaa ei ole. Englantilaiset käyttävät kehittimistä sanaa shell (kuori), joka kuvaa hyvin niiden luonnetta.

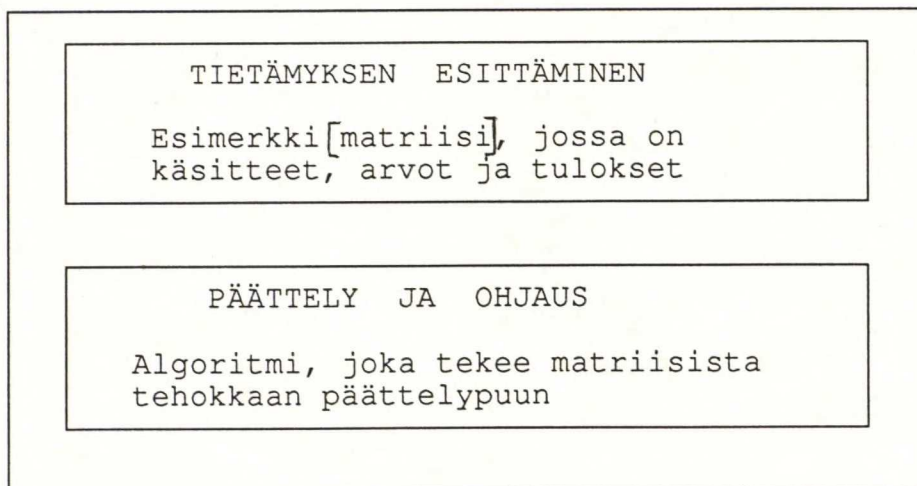


kuva 13. Asiantuntijajärjestelmien rakentamisvälineet /7/

## 4.2 Kehittimien luokkajako

### 4.21 Induktiiviset kehittimet

Induktiiviset kehittimet tekevät sääntöjä käyttäjän syöttämistä esimerkeistä; kehittimen algoritmi tekee säännöt tai yksinkertaisen päättelypuun niistä. Pääöstä tehdessään kehitin siirtää esimerkit päättelypuuhun ja prosessin aikana se priorisoi käyttäjälle esitettävien kysymysten järjestyksen. Induktiiviset kehittimet ovat helppokäyttöisiä, mutta melko joustamattomia ja tehottomia työkaluja. Ne ovat kuitenkin erittäin käyttökelpoisia ongelmiin, joista on lukuisia oikeisiin lopputuloksiin johtaneiden tapausten esimerkkejä. Nämä työkalut ovat matriisipohjaisia ja niitä on saatavissa sekä PC-, minikone-, että mainframeversioina. Induktiiviset kehittimet ovat helppokäyttöisiä ja ne vähentävät olennaisesti tietämyksen hankintaan kuluva-aikaa. Ne soveltuvat parhaiten yksinkertaisiin esimerkkeihin perustuvien ongelmien ratkaisuihin. Joihinkin sääntöpohjaisiin kehittämiin on saatavissa induktioon perustuva sääntögeneraattori erillisenä ohjelmana. /7/



kuva 14. Induktiivisen kehittimen rakenne /7/

#### 4.22 Yksinkertaiset sääntöpohjaiset kehittimet

Yksinkertaiset sääntöpohjaiset kehittimet ovat PC:lle tehtyjä, 'IF-THEN' sääntöjä käyttäviä työkaluja. Useimmat niistä käyttävät taaksepäin päättelyä. Ne soveltuvat pienehköjen, alle 500 säännön aj:n rakentamiseen. Yksinkertaiset sääntöpohjaiset kehittimet eroavat rakenteellisista sääntöpohjaisista kehittimistä mm. siten, että niissä ei ole 'rakennepuuta' (context tree), eikä yhtä hyviä editointimahdollisuuksia ja niiden tietämys sijaitsee yhdessä tietämyskannassa. Ne eivät pyri esittämään todellista asiantuntemusta, vaan useimmiten ne tarjoavat käyttäjälle erikoistuneita neuvoja pienehköissä mutta vaikeissa ongelmissa. Ne käyttävät tietämystekniikan menetelmiä, mutta soveltuvat silti lähinnä proseduurimaisen tietämyksen esittämiseen. Näiden kehittimien avulla voi ohjelmointiin tottumatonkin käyttäjä rakentaa tehokkaita järjestelmiä, joita voi ajan myötä täydentää vaatimusten kasvaessa. Yksinkertaiset sääntöpohjaiset kehittimet soveltuvat erittäin hyvin 'epärakenteellisten' ongelmien ratkomiseen. /7/

#### 4.23 Rakenteelliset sääntöpohjaiset kehittimet

Rakenteelliset sääntöpohjaiset kehittimet tarjoavat yleensä rakennepuun, varmuuskertoimet, hyvät editointimahdollisuudet ym. tehokkaita piirteitä. Suuret sääntöpohjaiset työkalut toimivat yleensä suurissa tietokoneissa kuten VAX:ssa, LISP-koneissa tai UNIX-työasemissa. PC:ssä toimivissa keskikokoisissa kehittimissä on myös rakennepuut. Nämä kehittimet käyttävät 'IF-THEN' sääntöjä, jotka voidaan jakaa rakennepuuhun hierarkisesti järjestettyihin 'paketteihin'. Tällaiset 'sääntöpaketit' toimivat kuin itsenäiset tietokannat, jotka saavat lisää informaatiota, kun toisia



sääntöpaketteja on tulkittu. Nämä järjestelmät ovat parhaimmillaan tapauksissa, joissa on suuri määrä sääntöjä, jotka voidaan jakaa alijoukkoihin. Keskikokoiset PC-pohjaiset kehittimet pystyvät käyttämään ulkopuolisia tietokantojakia. Juuri tämä ominaisuus erottaa nämä kehittimet pienistä kehittimistä. /7, 23/

#### 4.24 Hybridikehittimet

Hybridikehittimet edustavat monimutkaisinta aj-kehitysympäristöä, mitä saatavissa on. Viime aikoihin asti hybridityökaluja on ollut saatavissa ainoastaan LISP:llä toimivina VAX:hin, LISP:lle tehtyihin UNIX-työasemiin ja LISP-koneisiin. Keskikokoiset hybridityökalut ovat uusi työkaluluokka, jotka tarjoavat useita tietämyksen esittämisvälineitä PC:lle. Niiden suosion odotetaan kasvavan nopeasti, kunhan PC:n kapasiteetti paranee ja ihmiset huomaavat, että PC:llä pystyy rakentamaan ja käyttämään myös suuria asiantuntijajärjestelmiä. Nämä kehittimet käyttävät objektiorientoitunutta ohjelmointitekniikkaa esittämään ongelman elementtejä, joita järjestelmä käsittelee objekteina. Objektit voivat sisältää tosiasioita, IF-THEN sääntöjä tai osoittimia muihin objekteihin. Nämä työkalut ovat vaikeita käyttää ja vaativat käyttäjältä hyvää LISP:in ymmärtämistä sekä hyvää tuntemusta tietokoneesta, jolla sovellusta ajetaan. Hybridikehittimet soveltuvat suurille, yli 500 sääntöä sisältäville järjestelmille. Niissä on yleensä graafinen käyttäjäliityntä. Hybridityökalut ovat erittäin tehokkaita, mutta niin vaikeita ja kalliita rakentaa, että ainoastaan ongelman ollessa mutkikas ja suuri on niiden käyttö järkevää. Poikkeuksia ovat mikrotietokonepohjaiset hybridikehittimet. Nykyään niitä käytetään lähinnä tutkimuksessa, mutta tulevaisuudessa ne leviävät varmastikin myös käytännön sovelluksiin. /7, 23/

#### 4.25 Sovellusaluekehittimet

Sovellusaluekehittimet (domain specific tools) on suunniteltu tietyn erikoisalan aj:en kehittämiseen. Ne voivat käyttää mitä tahansa edellä mainittuja tekniikoita ja ne voidaan luokitella mihin tahansa em. luokkaan. Nämä työkalut tarjoavat erikoiset kehittäjä- ja käyttäjäliitynnät, jotka mahdollistavat sen alan aj:n rakentamisen paljon nopeammin kuin muilla työkaluilla. Tällaisten aj-kehittimien suosion odotetaan kasvavan nopeasti lähitulevaisuudessa. /7/

## 5. TIETEKNIIKAN ASIANTUNTIJAJÄRJESTELMIÄ

### 5.1 CHINA

CHINA (Computerized HIGHway Noise Analyst) sääntöpohjainen asiantuntijajärjestelmä maanteiden melusteiden suunnitteluun. Sen on kehittänyt kolme tieinsinööriä, joilla on vankka kokemus melusteiden suunnittelusta. CHINAN alkupe-  
räinen prototyyppi tehtiin LISP-pohjaisella kehittimellä, Geniellä ja se toimi VAX:n minitietokoneessa. CHINA on tyyppillinen esimerkki nykyaikaisen asiantuntijajärjestelmän käytöstä tietekniikassa : vaikea suunnittelutehtävä, joka vaatii runsasta käytännön kokemusta, vankkaa asiantuntemusta ja monimutkaista laskemista. Laskentaosuus CHINAssa on hoidettu liittymällä ulkopuoliseen Fortran-melulaskentaohjelmaan. /3/

Melusteiden akustisessa suunnittelussa Yhdysvalloissa käytetään mm. kahta algoritmista Fortran-ohjelmaa STAMINA:a ja OPTIMA:a. STAMINA-ohjelmalla lasketaan meluarvot halutuista mittauspisteistä tietyillä alkuarvoilla (mm. autojen määrä, nopeus ja melusteen korkeus). OPTIMA käyttää tätä 'meludataa' iteratiivisessa hinta-hyötysuhdeanalyysissä, jossa tekijöinä ovat mm. melusteen materiaali ja sallitut meluarvot. OPTIMA tulostaa taulukon hinta-hyötysuhteista, jonka tulkitseminen ja sen avulla suunnittelu vaatii kokemusta. Tältä pohjalta päätettiin kehittää asiantuntijajärjestelmä, joka auttaa kokematontakin suunnittelijaa tekemään korkealaatuisia melustesuunnitelmia. /3/

Alustavissa kokouksissa tutkittiin aj-tekniikan soveltuvuus tähän tehtävään ja kartoitettiin ongelmia, joihin tieinsinööri törmää käyttäessään OPTIMA:a melusteiden suunnittelussa. Kokouksissa kukin asiantuntija selitti oman suunnittelufilosofiansa ja suunnittelun eri vaiheet kirjat-



tiin. Ristiriidat ja tapaukset, joissa yhteisymmärrykseen päästiin kirjattiin myös. Ristiriita-tapauksista neuvoteltiin myöhemmin ja pyrittiin niissäkin pääsemään kaikkia asiantuntijoita tyydyttävään ratkaisuun. Näiden neuvottelujen tuloksena saatiin menettelytapa, joka käytti OPTIMA:a kolmen todellisen asiantuntijan taidoilla./3/

Kun yhteisymmärrykseen suunnittelumenetelmistä oli päästy, päätettiin tehdä kirjallisuustutkimus, joka keskittyi kahteen seikkaan : 1) etsitään kaikki OPTIMA:n käytöstä kertyneet kokemukset ja 2) tutkitaan kaikki menetelmät, joissa asiantuntijajärjestelmään on liitetty ulkopuolisia analyttisiä ohjelmia. /3/

Valittavalle kehittimelle asetettiin useita vaatimuksia, joista tärkein oli kehittimen helppokäyttöisyys. Myös järjestelmän muokattavuutta, eli ohjelman muuntelua ja uusien ominaisuuksien lisäämistä, pidettiin tärkeänä ominaisuutena kehitintä valittaessa. Koska laskentaosuus oli päätetty säilyttää OPTIMA:n muodossa, oli kehittämissä oltava myös liittynät ulkopuolisiin Fortran-ohjelmiin. Kuten edellä mainittiin, alkuperäisen prototyypin tekemisessä käytettiin LISP-pohjaista kehitintä ja VAX:in minikonetta. Myöhempiä versioita on tehty niin MICRO-VAX:ille kuin IBM-yhteensopiville mikroillekin, kun GENIEstä tehtiin mikroversio Golden Common Lispillä./3/

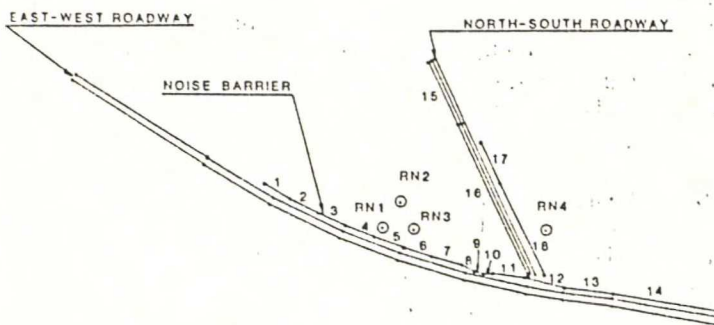
Kommunikointi muiden asiantuntijoiden kanssa kehitysprosessin aikana on hyvin tärkeää. Näin saadaan useampia näkökohtia ongelman tarkasteluun lopullisessa järjestelmässä. CHINA:n tapauksessa ulkopuolisten asiantuntijoiden kanssa käytiin keskusteluja komiteassa, joka tutkii liikenteen aiheuttamia meluhaittoja (Transportation Research Board's Committee on Transportation-Related Noise and Vibration). Tässä komiteassa ovat jäseninä jokaisen osavaltion tielai-

toksen asiantuntijoita sekä konsulttitoimistojen ja yliopistojen asiantuntijoita.

CHINAN kehittäjät käyttivät prototyyppilähestymistapaa ja he pyrkivät rakentamaan ensimmäiset versiot mahdollisimman pian. Sääntöjen ohjelmointi aloitettiin heti, kun oli päätetty, että asiantuntijajärjestelmä rakennetaan. Testaaminen aloitettiin yksinkertaisilla tapauksilla, joissa asiantuntijat olivat yhtä mieltä ratkaisusta. Näillä pyrittiin tuomaan esiin mahdolliset suuret virheet ja epäjohdonmukaisuudet. Jatkossa tutkittiin mutkikkaampia ongelmia, jotta yksityiskohtien pienimmätkin virheet saatiin selville. Seuraavassa esitellään kaksi testitapausta. /3, 8/

### 5.11 Testitapaus 1.

Kuvassa 15. on alue, jota tutkittiin. Alueella on neljä melunlaskentapistettä (vastaanotinta), joista yksi sijaitsee päätien (north-south-roadway) toisella puolella, erillään muista vastaanottimista. Meluesteet, jotka on merkitty 15-18, ovat todellisuudessa maaleikkauksen yläreunaa, joka käytännössä toimii meluesteenä. /8/



Kuva 15. Testitapaus 1 /8/

Suunnittelija haluaa tutkia maantien rakentamisen vaikutukset laskentapistessä 4 (vastaanotin RCN 4), mutta tietää

kokemuksesta, että meluesteen rakentaminen ei ole hinta-hyötysuhteen kannalta järkevää. CHINA kohtaa kaksi ongelmaa tällä kohtaa. Ensimmäinen on se, kuinka yhtä laskentapistettä (RCN 4) tulisi käsitellä esteiden 8 - 14 osalta. Toinen on se, kuinka esteitä 15 - 18, tulisi käsitellä meluesteinä, kun nämä esteet ovat ainoastaan maaleikkauksen yläpintaa, jota ei koroteta iteratiivisisten laskentakierrosten aikana. Taulukossa 1. on ehdotukset tehtävistä meluesteistä ja taulukossa 2. on meluesteiden vaikutukset.  
/8/

Taulukko 1. Testitapaus 1. meluesteet /8/

| esteen<br>numero                          | esteen korkeus (m)                |                             |
|---|-----------------------------------|-----------------------------|
|   | asiantuntija,<br>hinta= \$157.100 | CHINA,<br>hinta = \$127.700 |
| 2   | 4.0                               | 3.4                         |
| 3   | 4.0                               | 3.4                         |
| 4   | 4.6                               | 4.0                         |
| 5   | 5.2                               | 4.0                         |
| 6   | 5.2                               | 4.0                         |
| 7   | 4.0                               | 4.0                         |
| Esteet 1 ja 8-18 asetettiin 0 m korkeiksi |                                   |                             |



Taulukko 2. Testitapaus 1. Melusteiden vaikutus /8/

| RCN   | L [Z(0)]<br>a) | L b)              |       | suojaus (dBA)     |       |
|---|----------------|-------------------|-------|-------------------|-------|
|   |                | asian-<br>tuntija | CHINA | asian-<br>tuntija | CHINA |
| RN1   | 72.5           | 64.8              | 65.5  | 7.8               | 7.1   |
| RN2   | 71.1           | 65.7              | 66.1  | 5.4               | 5.0   |
| RN3   | 72.1           | 65.7              | 66.2  | 6.4               | 5.9   |
| RN4   | 68.5           | 68.3              | 68.3  | 0.2               | 0.2   |
| a) melutaso ilman suojausta<br>b) melutaso suojauksen jälkeen |                |                   |       |                   |       |

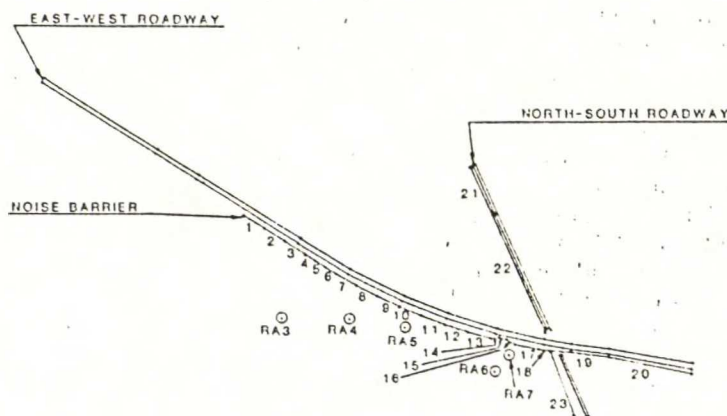
Tarkasteltaessa asiantuntijan ratkaisua huomataan, että tavoiteltu 5 dBA:n vähennys melutasossa saavutettiin \$157.000 hinnalla. Melusteet 1 ja 8-18 jätettiin pois, niiden merkityksen ollessa vähäinen.

CHINAN tulokset on esitetty samoissa taulukoissa. Myös CHINA jätti melusteet 1 ja 8-18 kokonaan pois. CHINA ei myöskään pyrkinyt nostamaan esteitä 15-18, jotka ovat maaleikkauksen yläpintaa. Sekä asiantuntija että CHINA jättivät laskentapiste RN4 huomiotta (0.4 dBA vaimennus verrattuna 5.0 dBA:n tavoitteeseen). Suunnitelman hinnaksi tuli n. 20% vähemmän kuin asiantuntijan suunnitelman. Hintaaero johtui pääasiassa siitä, että asiantuntija valitsi korkeammat esteet ja ylitti näin tavoitteeksi asetetun 5 dBA:n melun

vähennyksen. Tämä oli henkilökohtainen valinta, jota ei voi sanoa asiantuntemuksesta johtuvaksi seikaksi. CHINAN ehdotukset aiheuttivat kuitenkin sen, että asiantuntija muutti omia suunnitelmiaan siten, että suunnitelmien hinta aleni 4%. /8/

## 5.12 Testitapaus 2.

Kuvassa 16. on testitapaus 2. suunnitelma. Melunlaskentapisteitä on viisi, joista yksi, RA3 on erillään muista ja kauempana maantiestä kuin muut laskentapistet. Myös laskentapiste RA4 on hiukan erillään muista ja sijaitsee lähempänä maantietä ja mittaa siten korkeampia melutasoja. Tämä on jälleen tapaus, jossa asiantuntija saattaa olla ottamatta erillään olevat laskentapistet huomioon. Sekä asiantuntijan että CHINAN ratkaisut on esitetty taulukoissa 3 ja 4. /8/



kuva 16. Testitapaus 2./8/

Taulukko 3. Testitapaus 2. meluesteet /8/

| esteen<br>numero                                | esteen korkeus (m)                   |                                |
|---|--------------------------------------|--------------------------------|
|   | asiantuntija, a)<br>hinta= \$288.000 | CHINA, b)<br>hinta = \$258.800 |
| 7   | 5.2                                  |                                |
| 8   | 5.2                                  |                                |
| 9   | 5.2                                  | 4.0                            |
| 10  | 5.2                                  | 4.6                            |
| 11  | 5.2                                  | 4.0                            |
| 12  | 5.2                                  | 4.6                            |
| 13  | 5.2                                  | 4.6                            |
| 14  | 5.2                                  | 4.0                            |
| 15  | 5.2                                  | 4.0                            |
| 16  | 5.2                                  | 4.6                            |
| 17  | 5.2                                  | 5.2                            |
| 18  |                                      | 4.6                            |
| 19  |                                      | 3.4                            |
| a) Esteet 1-6 ja 18-23 asetettiin 0 m korkeiksi |                                      |                                |
| b) Esteet 1-8 ja 20-23 asetettiin 0 m korkeiksi |                                      |                                |



Taulukko 4. Testitapaus 2. Melusteiden vaikutus /8/

| RCN   | L [Z(0)]<br>a) | L b)              |       | suojaus (dBA)     |       |
|---|----------------|-------------------|-------|-------------------|-------|
|   |                | asian-<br>tuntija | CHINA | asian-<br>tuntija | CHINA |
| RA3   | 63.5           | 63.0              | 63.5  | 0.5               | 0.0   |
| RA4   | 67.5           | 64.4              | 67.2  | 3.2               | 0.4   |
| RA5   | 70.5           | 63.0              | 65.2  | 7.5               | 5.4   |
| RA6   | 68.1           | 63.1              | 63.2  | 5.0               | 5.0   |
| RA7   | 72.0           | 64.1              | 63.6  | 7.9               | 8.4   |
| a) Melutaso ilman suojausta<br>b) Melutaso suojauksen jälkeen |                |                   |       |                   |       |

Verrattaessa asiantuntijan ja CHINAN suunnitelmia huomataan, että asiantuntija jätti ottamatta huomioon laskentapisteen RA3 ja poisti melusteet 1-6. CHINA jätti ottamatta huomioon sekä laskentapisteen RA3 että laskentapisteen RA4. CHINA käsitteli jälleen hyvin esteet, jotka todellisuudessa ovat maaleikkauksen yläpintaa, yrittämättä kohottaa niiden reunaa. CHINAN suunnitelmat olivat jälleen halvempia, mutta ne eivät suojanneet laskentapisteitä niin hyvin melulta kuin asiantuntijan suunnitelmat. CHINAN suunnitelmissa laskentapisteiden RA6 ja RA7 meluvähennykset pärjäävät vertailussa

asiantuntijan suunnitelmiin, mutta RA5 kohdalla CHINA ei pystynyt samaan kuin asiantuntija. Tämä johtui siitä, että CHINA madalsi esteiden 7 ja 8 korkeutta, kun asiantuntija piti ne ennallaan. Nämä esteet suojasivat myös laskentapistettä RA5, mutta CHINAN tietämyksen mukaan suojaus ei ollut niin merkittävää, että se olisi kannattanut ottaa huomioon. Laskentapisteeessä RA4 puolestaan asiantuntijan suunnitelmat pärjäsivät huonommin kuin CHINAN suunnitelmat. Asiantuntijan ratkaisu pudotti melutasoa ainoastaan 3.2 dBA, joka on huomattavan vähän./8/

CHINAN ehdottamien meluesteiden yläpinnat ovat epätasaisia, mutta siitä huolimatta sen suosittalemaa ratkaisua on pidettävä hyväksyttävänä, ottaen huomioon CHINAN sen hetkisen tietämyskannan laajuuden./8/

## 5.2 SCEPTRE (Surface Condition Expert for Rehabilitation Planning)

SCEPTRE on Yhdysvalloissa kehitetty interaktiivinen mikrotietokonepohjainen asiantuntijajärjestelmä päällysteiden kunnostusstrategioiden suunnitteluun ja kehittämiseen. Se arvioi päällysteeseen kohdistuvat rasitukset sekä monia muita käyttäjän syötteitä tunnistaakseen ja suositellakseen sopivia kunnostus- ja kunnossapitostrategioita ( RAM, rehabilitation and maintenance strategies). SCEPTRE on rakennettu IBM-yhteensopivissa mikroissa toimivalla EXSYS-kehittimellä./21/

SCEPTRE kyselee käyttäjältä lähtötietoja, joita järjestelmä käyttää päättelyissään ja tehdessään päätelmiä, jotka perustuvat tietämyskantaan koodattuihin tietoihin ja heuristiikkoihin. Tietämyskanta on rakennettu kahden päällystealan asiantuntijan avustuksella. Tässä tarkasteltu versio 1.4 soveltuu osavaltion ylläpitämiin joustaviin päällysteisiin ja vastaa lähinnä Washingtonin osavaltiossa olevaa tilannetta. SCEPTREN parametrit voidaan kuitenkin asettaa sopiviksi erilaisiin olosuhteisiin. /21/

Tunnistaessaan sopivia kunnossapitostrategioita SCEPTRE harkitsee 24 vaihtoehtoista strategiaa. Käyttäjän syötteiden perusteella SCEPTRE muokkaa vaihtoehdot sopivaksi alijoukoksi. SCEPTRESSä on kymmenen perusstrategiaa :

- 1) do nothing
- 2) fill cracks
- 3) fog seal
- 4) friction course
- 5) chip seal
- 6) double chip seal
- 7) thin asphalt concrete overlay (thickness  $\leq$  3cm)
- 8) medium asphalt concrete overlay ( 3 - 7.5 cm)



- 9) thick asphalt concrete overlay (thickness > 7.5 cm)
- 10) mill and replace

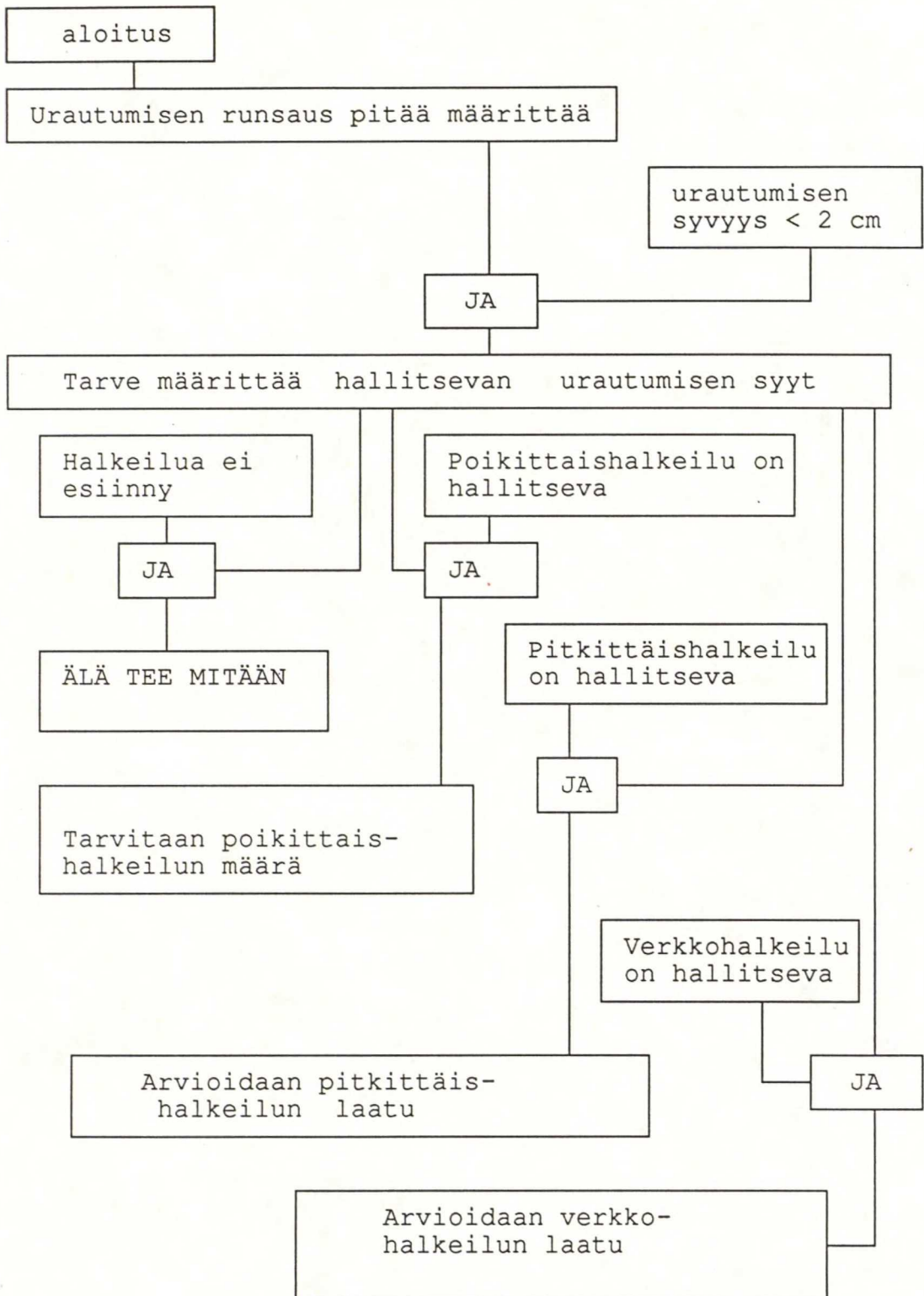
Jos päällysteessä on useita vauriotyyppejä samanaikaisesti tai vauriot ovat vakavia, näitä kymmentä perusstrategiaa muokataan siten, että ensin suositellaan päällysteen tasoittamista tai tasoittamista ja jauhamista osana päällysteen kunnostamistyötä. Nämä kymmenen perusstrategian eri muunnokset muodostavat 24 vaihtoehtoista kunnossapitostrategiaa. Päällysteen pinnan vauriot voivat olla

- 1) verkkohalkeamat pyöräurissa
- 2) urautuminen
- 3) pitkittäishalkeamat pyöräurissa
- 4) poikittaishalkeamat.

Nämä vauriotyypit ovat samat kuin mitä Washington State Department of Transportation (WSDOT) käyttää omassa päällysteiden kunnossapitojärjestelmässään (pavement management system, PMS).

Tietämyksen esittäminen

SCEPTRE on sääntöpohjainen aj, eli sen tietämys on esitetty 'JOS ehto NIIN suorita'-muotoisilla säännöillä. SCEPTRE käyttää taaksepäinpäätelyä.

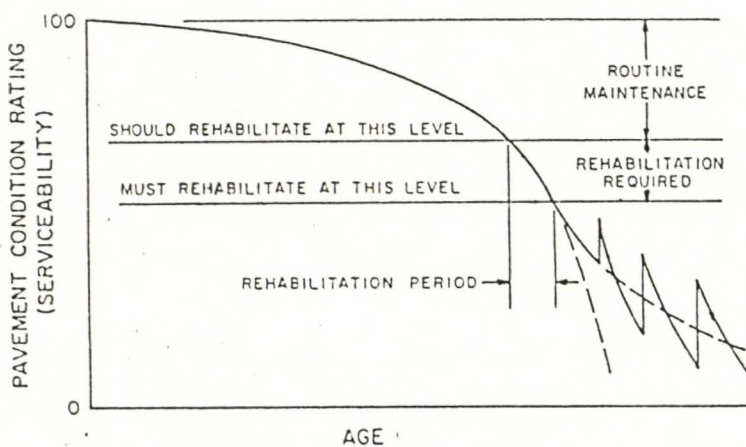


Kuva 17. SCEPTRE päättelyverkon osa /21/

SCEPTREN tietämuskannassa on 140 monimutkaista päättelysääntöä.

EXSYS-kehittämissä on liittynyt ulkopuolisiin ohjelmiin, joilla pystytään välittämään tietoa järjestelmästä ulos tai tuomaan uutta tietoa sisään. SCEPTRESSä tätä ominaisuutta on hyödynnetty sisällyttämällä kunnossapitostrategioiden kustannushyötyanalyysi järjestelmään. Fortranilla kirjoitettu ohjelma saa dataa valituista strategioista ja se suorittaa monimutkaiset laskelmat ja tulostaa lopputulokset./21/

Jokaisen kunnossapitostrategia-analyysin taustalla on päällysteen 'elinikä-käyrä' (kuva 18). Tämä käyrä kuvaa päällysteen ikä/kunto suhdetta. Päällysteen kunto arvioidaan funktiolla, jossa yhdistetään ajomukavuus ja päällysteen kunnan arviointi. Päällysteen kunto arvostellaan asteikolla 0 - 100. Moitteeton päällyste saa arvon 100. Tässä WSDOT:n menetelmässä määritellään kaksi kunnostusrajaa: 'pitäisi kunnostaa' ja 'täytyy kunnostaa'. Numeroarvoina nämä rajat vastaavat 60 ja 40. Kun päällysteelle tehdään kunnostustoimenpide, sen kunto nousee jälleen arvoon 100.



Kuva 18. Päällysteen 'elin-ikä'-käyrä./21/



SCEPTRE tarjoaa käyttäjälle kaksi vaihtoehtoa esittää päällysteen elinikäkäyrä. Ensimmäinen on lineaarinen arviointi, jossa päällysteen kuntoarvo on 100 iässä 0 ja 'täytyy kunnostaa'-arvo 40, joka vastaa päällysteen palveluikää sekä WSDOT:n kurvilineaariset funktiot, jotka vaihtelevat kunnostustoimenpiteen mukaan.

SCEPTRE:n versiossa 1.4 on tällä hetkellä vain lineaarinen arviointi. Asiantuntijat ovat arvioineet kunnostustoimenpiteiden palveluiät ja nämä tiedot ovat järjestelmän tietämyskannassa. Näistä palvelui'istä johdetaan jokaisen kunnostustoimenpiteen suorituskäyrä. Syyt miksi päällysteille johdetaan nämä käyrät ovat seuraavat: 1) se visualisoi käyttäjälle jokaisen kunnossapitotoimenpiteen vaikutuksen ja keston sekä 2) siitä voidaan tehdä suuntaa-antava kustannus-hyötyanalyysi joka käsittää sekä päällysteen toimivuuden sekä kustannukset koko sen eliniälle. Tässä tapauksessa päällysteen hyväksyttävä kunto on alueella, joka jää suorituskäyrän ja arvon 40 väliin. Mitä suurempi tämä alue on sitä parempi on päällysteen kestävyys. Kunkin kunnossapitotoimenpiteen elinikä-kustannukset on laskettu vuoden 1986 keskimääräisten WSDOT:n projektikustannusten mukaan, ja niihin kuuluvat sekä rakennus- että ylläpitokustannukset. Elinikä-kustannukset muutetaan vuosittaisiksi kustannuksiksi kaistaa kohti, jotta erilaisten toimenpiteiden kustannuksia voidaan vertailla.

### 5.3 Asiantuntijajärjestelmä maantien ja rautatien risteyksien turvallisuuden parantamiseksi

Yhdysvalloissa kuoli vuosina 1977-1983 3067 ihmistä auton ja junan yhteentörmäyksissä tasoristeyksissä. Tämän vuoksi liittovaltio ryhtyi rahoittamaan risteysten turvallisuutta parantavia toimenpiteitä. Jotta varat käytettäisiin mahdollisimman hyödyllisesti, päätettiin kehittää menetelmiä risteysten arvosteluun, jonka mukaan varat jaetaan. Perinteisellä menetelmällä risteysten vaarallisuus arvioitiin matemaattisilla ja algoritmisilla malleilla. Näillä malleilla pyrittiin arvioimaan kunkin risteyksen onnettomuusalttius tulevana vuosina ottamalla huomioon kaikki tilastollisesti tärkeät tekijät. Vaikka malleilla pystyttiin ennustamaan onnettomuuksien lukumäärät kohtalaisen tarkasti, niissä ei kyetty ottamaan huomioon sosiaalisia ja poliittisia seikkoja, jotka vaikuttavat määrärahojen jakamiseen. Eri malleja tutkittiin ja parhaimmaksi havaittiin DOT-malli (Hitz ja Cross). Siinä yhdistyy kaksi toisistaan riippumatonta ennustetta. Ensimmäinen on peruskaava, joka ennustaa onnettomuusmahdollisuutta risteyksen perusominaisuuksien mukaan. /4/ Kaava on seuraavanlainen :

$$a = K * EI * MT * DT * HP * MS * HT * HL \quad (1)$$

missä a = ennustetut onnettomuudet risteyksessä per vuosi

K = vakio

EI = juna- ja autoliikenteen määrä

MT = raiteiden lukumäärä

DT = junia päivässä valoisana aikana

HP = onko maantie päällystetty vai ei

MS = maksimi aikataulu nopeudelle

HT = maantien tyyppi

HL = maantien kaistojen lukumäärä.

Toinen kaava kuvaa risteyksessä tapahtuneita onnettomuuksia.

Kaavassa oletetaan, että risteyksessä tulee tapahtumaan keskimäärin yhtä paljon onnettomuuksia vuodessa kuin aikaisemminkin./4/

$$A = \frac{T^0}{T^0 + T} (a) + \frac{T}{T^0 + T} \left( \frac{N}{T} \right) \quad (2)$$

missä A = lopullinen onnettomuusennuste  
 N/T = onnettomuus historia ennuste, onn./vuosi  
 N = tapahtuneiden onnettomuuksien lukumäärä  
 T = risteystä tarkkaillut vuodet  
 $T^0$  = kaavan painotuskerroin  $1/(0.05 + a)$

Vaikka tämä malli ottaa huomioon tärkeimmät tilastolliset tekijät, se ei ole riittävä, kun määrärahoja jaetaan koko osavaltioon. Käytännön arvostelussa kaavaa käytetään yhtenä valitsemiskriteerinä. Arvostelua täydennetään havainnoilla paikan päällä sekä monilla muilla laadullisilla tekijöillä, joita on mahdotonta sisällyttää matemaattiseen malliin. Lopullinen risteysten arvostelu koostuu siis sekä onnettomuusennusteesta että monista nyrkkisäännöistä.

Asiantuntijajärjestelmä päätettiin rakentaa, jotta risteysten luokittelu pystyttäisiin hoitamaan tehokkaasti sekä siksi, että on ennustettu kolmanneksen tämän hetken asiantuntijoista siirtyvän eläkkeelle seuraavan vuosikymmenen aikana. /4/

Ensimmäinen tehtävä oli hahmottaa ongelman ratkaisun tärkeimmät tekijät. Näitä ovat mm. ketkä osallistuvat järjestelmän kehittämiseen, ongelman luonteen analysointi, resurssit ja tavoitteet.



Ennen kuin tietämyksen hankintaan ryhdyttiin, osanottajat valittiin ja tehtävät jaettiin. Asiantuntijaksi valittiin Rail and Public Transportation Division of the Virginia Department of Transportation:in työntekijä, joka oli käytännössä joutunut arvioimaan maanteiden ja rautateiden tasoristeyksiä. Hänen tehtävänään oli käydä läpi koko valintaprosessi ja selvittää tietämysinsinööreille ongelman rakenne. /4/

Asiantuntijajärjestelmän tehtävänä on luokitella risteykset niiden vaarallisuuden mukaan. Järjestelmä käyttää luokittelussa sitä asiantuntemusta, mikä sen tietämyskantaan on koodattu tulkitsemaan lähtötietoina syötettyjä DOT-mallin tuloksia. Kun DOT-mallin onnettomuustodennäköisyys on saatu, asiantuntijajärjestelmä tutkii, onko risteyksen tiedoissa muita riskitekijöitä, kuten huono näkyvyys tai vilkas jalankulkuliikenne. Yhdistämällä nämä tiedot asiantuntijajärjestelmä arvioi risteyksen sijoituksen vaarallisuuslistalla ja suosittelee sopivia parannustoimenpiteitä. Kuolemaan johtaneiden onnettomuuksien vaikutus risteyksen arviointiin on tyypillinen ei numeerisesti mitattava suure. Risteys saattaa olla teknisesti hyvä, mutta kuolemaanjohtaneen onnettomuuden sosiaalinen ja poliittinen vaikutus, eli ihmisten vaatimukset parantaa turvallisuutta juuri siinä risteyksessä, nostaa risteyksen sijoitusta vaarallisuuslistalla. /4/

The Rail and Public Transportation Division kerää vuosittain tiedot osavaltion jokaisesta maantien ja rautatien tasoristeyksestä. Tämä helpottaa asiantuntijajärjestelmän ylläpi-toa. Tietämyskannan tietämys voidaan pitää ajantasalla vähällä vaivalla, kun jokaisen risteyksen uudet tiedot ovat saatavissa samasta paikasta. /4/

Järjestelmään syötetään kahdenlaista dataa:

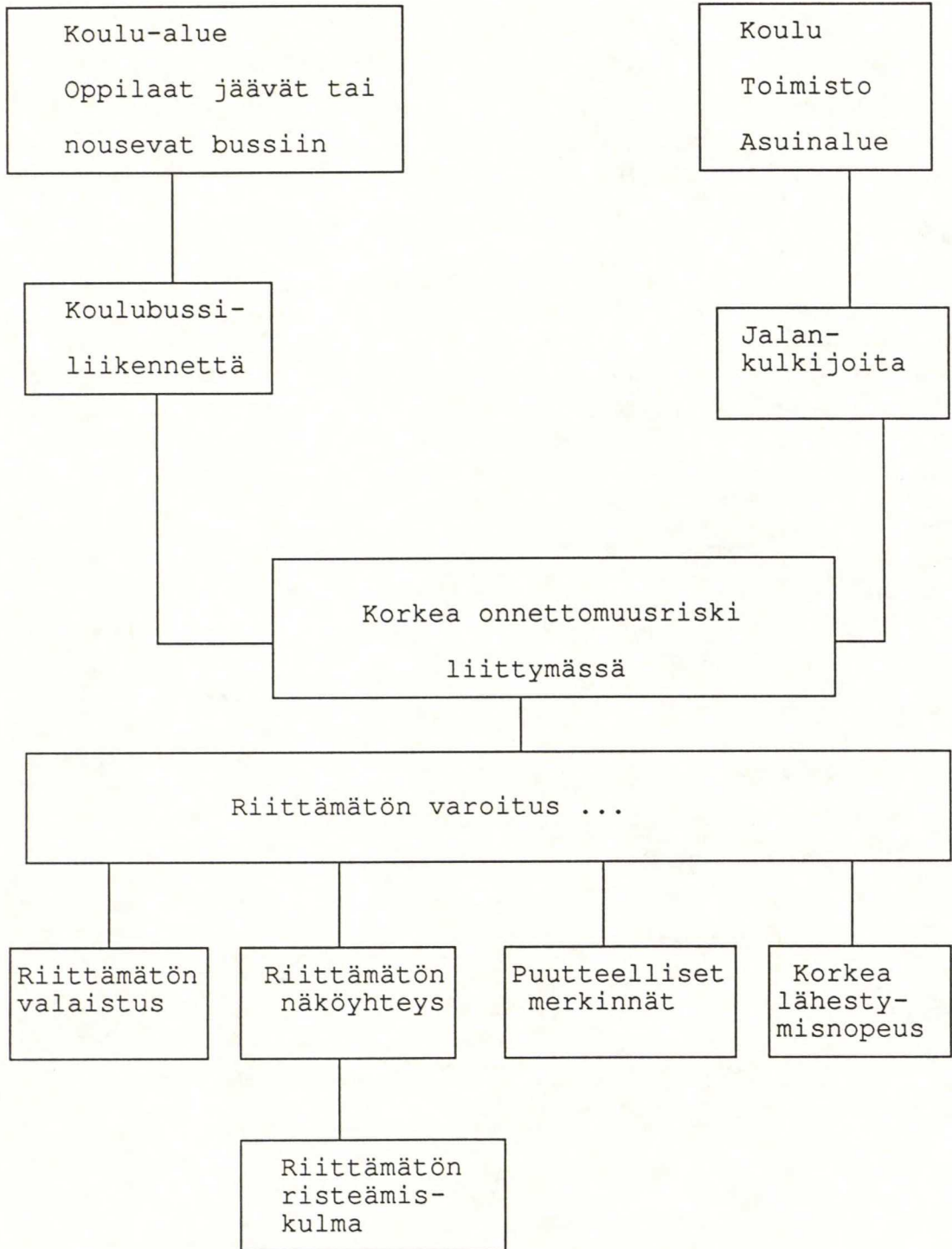
1) Numeerista tietoa, joka koostuu DOT-mallin onnettomuustodennäköisyydestä, näköetäisyydestä, liikenteen lähestymisnopeudesta ja risteyksen ylittävien koulubussien päivittäisestä lukumäärästä.

2) Laadullista tietoa, kuten onko kouluja tai työpaikkoja risteyksen lähellä, onko näkyvyys risteyksessä hyvä yms.

Seuraavien tekijöiden välinen yhteys on tunnistettu

- \* riittämätön näköyhteys ja risteyksen vaarallisuus
- \* runsas koulubussiliikenne ja onnettomuuden vakavuus
- \* radan ja päällysteen kunto risteyksessä ja kohonnut riski jalankulkijoille, pyöräilijöille ja muille kaksipyöräisten ajoneuvojen kuljettajille
- \* työpaikkojen ja ostoskeskusten sijainti ja runsas jalankulkuliikenne

Eri tekijöiden välisiä suhteita selventää kuva 19.



Kuva 19. Eri tekijöiden väliset suhteet./4/



Järjestelmä tulostaa lopputuloksena listan kaikista Virginian osavaltion yleisistä tasoristeyksistä. Lista on järjestetty 'vaarallisuusjärjestyksen' mukaan, ottaen huomioon sekä laadulliset että määrälliset seikat. Listassa on myös sopivat parannusehdotukset kullekin risteykselle. Kun kaikki tarpeellinen tieto jokaisesta risteyksestä on hankittu, järjestelmä tarkistaa ja korjaa alustavan prioriteettilistan tietämyskannassa olevan tietämyksen mukaan. Ongelman ratkaisu koostuu seuraavista päävaiheista:/4/

- 1) Hankitaan kaikki tarpeellinen tieto risteyksestä. Tieto siirretään järjestelmään vastaamalla sen esittämiin kysymyksiin.
- 2) Järjestelmä vastaa käyttäjän kysymyksiin ja auttaa selittämällä esim. miksi jokin tieto kysyttiin.
- 3) Selvitetään muuttujien väliset suhteet ja tallennetaan ne tietokoneen muistiin. Kaikki tärkeä laadullinen (kvalitatiivinen) ja määrällinen (kvantitatiivinen) tieto jokaisesta risteyksestä on järjestelmän muistissa.
- 4) Laaditaan lopullinen prioriteettilista. Alkuperäinen DOT-lista muokataan uuteen järjestykseen, kun kaikki mahdolliset tekijät on otettu huomioon.

Tämän asiantuntijajärjestelmän prototyyppi kehitettiin kahdella ohjelmointikielellä : TURBOPASCALilla ja GCLISP:llä (Golden Common LISP). TURBOPASCAL valittiin ongelman sisältyvän laskennallisen osuuden vuoksi.

Seuraava vaihe oli kerätyn tietämyksen formalisointi ohjelmointikielille sopivaan muotoon. Lukuisten haastat-

telujen ja keskustelujen asiantuntijoiden kanssa päätettiin edetä seuraavalla tavalla./4/

- 1) Tunnistetaan kaikki tekijät, jotka asiantuntija ottaa huomioon arvioidessaan risteyksen vaarallisuutta.  
(Näitä tekijöitä ei ole huomioitu DOT-mallissa.)
- 2) Laitetaan nämä tekijät tärkeysjärjestykseen.
- 3) Tutkitaan kunkin tekijän vaikutus risteyksen riskitasoon.

Tässä on lueteltu tärkeysjärjestyksessä ne tekijät, joilla on vaikutusta risteyksen vaarallisuuteen.

\* Näkyvyysarvot (Sight Distance Values)

Näkyvyys on etäisyys, jonka kuljettaja näkee tietä eteenpäin. Se vaikuttaa myös nopeuteen. Maantien ja rautatien risteyksessä määritellään kaksi näkyvyyttä: a) Kuinka paljon maantietä on näkyvissä ja b) kuinka paljon rautatietä on näkyvissä kuljettajalle.

\* Koulubussit (School Bus)

Jos koulubusseja kulkee rautatien yli useammin kuin kaksi kertaa vuorokaudessa, risteyksen vaarallisuus-indeksi nousee.

\* Vaarallisten aineiden kuljetukset (Hazardous Material Carriers)

Rautatieyhtiöt eivät ilmoita liikenneviranomaisille rautateillä tapahtuvista vaarallisten aineiden kuljetuksista. Ainoa saatavissa oleva tieto saatiin rekkakuljetuksista. Vaarallisten aineiden kuljetuksilla on sama vaikutus 'vaarallisuus-indeksiin' kuin koulubusseilla.

\* Vaarallisuus (Fatality)

Kuolemaan johtaneiden onnettomuuksien lukumäärä nostaa vaarallisuus-indeksiä. Itsemurhia tai rattijuopumuksesta aiheutuneita onnettomuuksia ei oteta huomioon.

\* Lainvalvonta (Law Enforcement)

Eri kunnissa lainvalvonnan tarkkuus vaihtelee sen mukaan, kuinka paljonvaroja on käytettävissä. Tällä voi olla vaikutusta risteysten vaarallisuuteen, joten puutteelliset lainvalvontaolot pyritään ottamaan huomioon prioriteettilistaa tehtäessä.

\* Vanhusten ja vammaisten hoitopaikkojen sijainti tasoristeyksen lähellä (Senior Citizen Centers or Centers for the Handicapped in the Vicinity of the Crossing)

Hoitopaikkojen sijainti risteysten lähetyvillä kaipaa tarkempaa tutkimista, koska vanhusten liikkuminen ja reaktiot ovat hitaita.

\* Jalankulkijoiden määrä (Pedestrian Activity)

Jalankulkijoiden määrä riippuu siitä, kuinka lähellä risteys on työpaikkoja tai asuinalueita. Jos asukkaiden tai työntekijöiden lukumäärä ylittää 300, sitä pidetään merkittävänä. Merkittävää jalankulkijoiden lukumäärää pidetään vaarallisena, jos työpaikka tai asuinalue on 12 m säteellä risteyksestä.

\* Liikennemerkkien näkyvyys pimeässä ( Night Visibility of Highway Signs)

Kylttien näkyvyyttä pimeässä voidaan pitää vaaratekijänä. Puut tai valaistut kyltit saattavat sokaista ajajan pimeässä.



## 5.4 Muita tietekniikan asiantuntijajärjestelmiä

### 5.41 ROSE

ROSE on asiantuntijajärjestelmä maanteiden ennalta ehkäisevään kunnossapitoon. Se on keskittynyt asfalttibetonin pinnan vaurioiden korjaamisen (routing and sealing) käyttöön kunnossapidossa erikoisesti kylmillä alueilla. Järjestelmän suositukset annetaan sillä perusteella, että pinnan jyrshintä ja pintausta on sopiva ja edullinen kunnossapitotoimenpide. ROSE on alunperin suunniteltu Ontarion liikenneministeriön (Ministry of Transportation and Communications) käyttöön. Pinnan vaurioiden korjaamisen tarkoituksena on estää pintavesien, erityisesti suolapitoisten vesien, pääseminen tien rakenteisiin ja estää tien rakenteen vaurioituminen.

ROSEN perustana on MTC:n tarkkailu- ja arviointijärjestelmä, ja siinä on liityntä päällysteiden ylläpitojärjestelmän rekisteriin (pavement management information data bank). Tietämyskanta, joka sisältää päättelylogiikan siitä, milloin pitää jyrsiä ja tiivistää, kehitettiin uusimpien MTC:n tutkimustulosten pohjalta. Järjestelmän tietämyskannassa on 360 sääntöä ja 26 numeerista muuttujaa, joilla kuvataan mm. päällysteeseen kohdistuva rasitus. Järjestelmä rakennettiin mikropohjaisella EXSYS-kehittimellä. ROSEsta tehtiin automaattinen versio Fortranilla, jota käytettiin n. 900 päällysteosuuden, eli n. 7200 maantie kilometrin arviointiin. Sovellus arvioi erilaisten jyrsimis- ja pintaustoimenpiteiden seuraukset./27/

ROSE on käytössä oleva prototyyppi, jota on testattu useisiin erilaisiin päällysteen ylläpidon ongelmiin. Kehittäjät arvioivat kuitenkin, että järjestelmän käyttö muissa olosuhteissa kuin mihin se kehitettiin, on vaikeaa ja jopa harkitsematonta vaikka metodologia ja yleinen lähestymistapa ovat yleisesti sovellettavissa./27/

#### 5.42 BETONIPÄÄLLYSTEEN ARVIOIJA

Tämä prototyyppi järjestelmä on kehitetty avustamaan tieinsinööriä betonipäällysteiden arvioinnissa. Monipuolinen päällysteen arviointi vaatii niin monen tekijän huomioon ottamista, että todellinen päällystealan asiantuntija suoriutuu tehtävästä parhaiten. Tämän järjestelmän tietämyskantaa luotaessa oli asiantuntijoina kaksi päällysteinsinööriä./27/

Järjestelmä rakennettiin Insight 2+ kehittimellä, ja se toimii IBM yhteensopivissa mikrotietokoneissa. Järjestelmä käyttää insinöörin keräämää tietoa selvittääkseen, mistä päällysteessä olevat vauriot ovat aiheutuneet sekä mitä kunnostustoimenpiteitä päällysteelle pitää tehdä. Järjestelmän laajuus oli aluksi rajattu saumallisen teräsbetonin käyttöön (jointed reinforced concrete), vaikka järjestelmän kehittäminen sopivaksi jatkuvaraudotteisen betonipäällysteen kanssa käytettäväksi onkin käynnissä. Järjestelmä on kehitetty Illinois'n yliopistossa (University of Illinois, Department of Civil Engineering)./27/

#### 5.43 STREET SMART

STREET-SMART on Kalifornian Yliopistossa (University of California, Irvine) kehitetty asiantuntijajärjestelmä, joka avustaa älykkäästi "Streets of the City" simulointi ohjelman käyttäjiä. Järjestelmä on tarkoitettu liikenneinsinöörien koulutukseen. Vaikka järjestelmä ei varsinaisesti olekaan tietekniikan asiantuntijajärjestelmä, se antaa hyvän esimerkin kuinka aj-tekniikkaa voidaan käyttää insinöörien koulutuksessa."Streets of the City" simuloi keskikokoisen



keskilännessä sijaitsevan kaupungin liikenneinsinöörin päätöksentekoja. Ohjelman käyttäjä pyrkii saavuttamaan katujen ja läpikulkuliikenteen osalta kaupungin johdon asetettamat tavoitteet kymmenen vuoden ajanjakson aikana. Jos vuosittainen toiminta on epäpätevää, niin liikenneinsinööri erotetaan virastaan ja simulointi lopetetaan. Simulointi on erittäin monimutkaista ja ohjelman käyttäjältä vaaditaan monimutkaisia päätöksentekoja joihin liittyy vaikeasti tulkittavia lehmänkauppoja. Suurin osa käyttäjistä saa potkut aikaisessa vaiheessa kymmenen vuoden jaksosta, mutta oppivat vähitellen ja parantavat suorituksiaan. Street Smart kehitettiin, kun tutkittiin voidaanko oppimista tehostaa "älykkäällä neuvojalla"./27/

Street Smartin tietämyskanta koostuu kahdesta osasta :

1) Semanttinen verkko, jossa jokainen solmu edustaa simuloinnin yhtä osaa, esim. liikenneturvallisuus indeksi, tai joukkoliikenteen ylläpito budjetti (Transit Maintenance Budget). Linkit eri solmujen välillä kuvaavat eri solmujen välisiä suhteita. Bussi kaluston ikä vaikuttaa bussien käytön hukka-aikaan, joka puolestaan vaikuttaa palvelun myöhästymis-indeksiin jne. Street Smart tietämyskanta sisältää 50 solmua ja 100 linkkiä perusverkkorakenteeseen./27/

2) Säännöt, jotka ohjaavat päättelystrategioita. Säännöt puuttuvat ensimmäiseksi niihin suoritusta mittaaviin kohtiin, jotka ovat vaikeimpia ja aiheuttavat usein suorituksen hylkäämisen. Sen jälkeen ehdotettu päätös tunnistetaan käytössä olevien tietojen pohjalta. Tämä ehdotus esitetään käyttäjälle, joka voi hyväksyä sen kokonaan tai osittain ja ottaa se mukaan seuraavan vuoden budjetin suunnitteluun.

STREET SMART on tehty muLISPillä mikrotietokeelle. Päätte-



lymekenismi käyttää eteenpäin päättelyä. Järjestelmä sisältää rajoitetun selitysmekanismin järjestelmän semanttisen verkon sisältämään tietoon kohdistuvista kysymyksistä. STREET SMARTia ei ole testattu laajasti, mutta on havaittu, että käyttäjät, joita ei avusteta harvoin suoriutuvat tehtävästään niin hyvin kuin ne, joita STREET SMART avustaa./27/

#### 5.44 OVERDRIVE

OVERDRIVE eli Overlay Design Heuristic Adviser, on asiantuntijajärjestelmä joustavien asfalttibetonipäällysteiden suunnitteluun olemassa olevien päällysteiden päälle. Yhdysvalloissa on viime vuosina käytetty paljon vanhan päällysteen päälle tehtäviä päällysteitä vähän tai kohdalaaisesti liikennöitävillä teillä. Yhdysvalloissa käytetään vuosittain miljardeja dollareita uudelleen päällystykseen (overlay) , joka oikein suunniteltuna voi olla erittäin edullinen keino päällysteiden puutteiden korjaamiseen huomattavan pitkiksi ajoiksi. OVERDRIVE on tarkoitettu, kuten em. SCEPTRE lähinnä kaupunkien ja piirikuntien insinöörien avuksi./27/

Alkuperäinen OVERDRIVEN prototyyppi perustui menetelmään, jota kutsutaan " component analysis overlay design method" eli pintauksen komponentti analyysi suunnittelu.

Tämä käsittää merkittäviä päätöksiä suunnitteluun vaikuttavien tekijöiden arvojen valinnassa. Jotkut näistä tekijöistä aiheuttavat 100 % vähennyksiä jokaisen päällyste kerroksen rakenteelliseen riittävyys ja voivat vaikuttaa ratkaisevasti selvitetessä uuden päällysteen tarvetta, sen paksuutta tai kustanuksia. OVERDRIVEN tietämyskanta on rakennettu päällystealan asiantuntijan ja kirjallisuudesta kerätyn tiedon yhdistelmänä. Järjestelmä on toteutettu EXSYS-kehittimellä ja se on sääntöpohjainen ja eteenpäinpäättelevä. Siinä on myös liitännät ulkopuolisiin ohjelmiin.

OVERDRIVEa kehitetään edelleen ja uusissa versioissa odotetaan olevan päällystealan asiantuntijoilta hankittua lisätietämystä./27/

#### 5.45 INTERSECTION ADVISOR

INTERSECTION ADVISOR on prototyyppi asiantuntijajärjestelmä, joka antaa parannusehdotuksia liittymien geometrian parantamiseksi. Päämääränä on kehittää asiantuntijajärjestelmä-lähestymistapa valo-ohjattujen liittymien suunnitteluun, niin että siihen sisältyy geometrinen tarkastelu. Nykyinen suunnittelu ei ota huomioon valo-ohjausta. Järjestelmä käsittelee myös tyypillisiä nelihaaraliittymiä (4 way intersection), jossa on kaksi läpimenevää kaistaa ja erilliset kaistat oikealle ja vasemmalle kääntyville./27/

Vuorovaikutteisen istunnon aikana järjestelmä pyytää tietoja risteyksen liikennemääristä, kriittisistä liikkeistä, geometriasta ja käytössä olevista kaistoista. Näistä tiedoista se pääättelee sopivimman ja tehokkaimman ratkaisun lähestymiskaistojen käytöstä. Järjestelmässä on 64 erilaista vaihtoehtoa ja 448 mahdollista suositusta jokaiselle analysoidulle vaihtoehdolle. Järjestelmään suunnitellaan parannuksia, kuten laajempaa tietämyskantaa ja liikennevalojen toiminnan vaiheistusta ja ajoitusta. Järjestelmän avulla tehdyt risteys suunnitelmat ovat pärjänneet hyvin vertailussa vuoden 1985 Highway Capacity Manual:in avulla tehtyihin risteys suunnitelmiin./27/

INTERSECTION ADVISOR on rakennettu M.1 kehittimellä ja se toimii IBM yhteensopivissa mikrotietokoneissa. Tämäkin järjestelmä on kehitetty yliopistossa (Department of Civil Engineering at North Carolina State University)./27/

#### 5.46 TIEN RAKENTEEEN SUUNNITTELUN TIETÄMYSJÄRJESTELMÄ

Teknillisessä Korkeakoulussa on tehty diplomityönä mikrotietokonepohjainen tietämysjärjestelmä tien rakenteen mitoittamiseen. Insight-2 kehittimellä tehty järjestelmä sisältää TVH:n suunnitteluohjeet ja tien rakenteen mitoittamiseen liittyvät laskentatehtävät. Järjestelmä on prototyyppi./14/



## 6. MIKROTIETOKONEPOHJAISEN ASiantuntijajärjestelmä- kehittimen valinta

### 6.1 Yleistä

Mikropohjaisten aj-kehittimien kaupallinen läpimurto on suhteellisen uusi asia tietotekniikan alalla. Sen vuoksi markkinat ovat melko kehittymättömät ja saatavissa olevien tuotteiden ominaisuudet vaihtelevat paljon. Sopivan tuotteen valitsemista vaikeuttaa tämän hetkisten tuotteiden lyhyt elinkaari. Uusia tuotteita ja uusia versioita tulee koko ajan, ja niiden laatu paranee jatkuvasti. On olemassa erilaisia mielipiteitä siitä, ovatko nykyiset mikrotietokonepohjaiset asiantuntijajärjestelmät todellisia asiantuntijajärjestelmiä vai ainoastaan uusia hiukan kehittyneempiä ohjelmointityökaluja. Todellisia asiantuntijajärjestelmiä näiden epäilijöiden mukaan olisivat vain suurissa tietokoneissa toimivat isot järjestelmät. Nykyiset mikrotietokonepohjaiset kehittimet käyttävät kuitenkin tietämystekniikan menetelmiä tietämyksen esittämiseen, joten niitä voitaneen kutsua aj-kehittimiksi, kun edessä oleva sana 'mikrotietokonepohjainen' selvittää niiden yleisen tason./15, 23/

Useimmissa mikropohjaisissa kehittimissä on yhdistelty eri tietämyksen esittämistekniikoita ja päättelymenetelmiä, joten tuotteiden luokittelu on vaikeaa. Eri tekniikoiden yhdistely antaa monipuolisemmat mahdollisuudet erilaisten ongelmien ratkaisuun. Vaatimukset käytettävälle laitteistolle ovat yleensä 640 kilotavua muistia sekä kovalevy. Valittaessa sopivaa kehittäintä tulee ottaa huomioon millaista tehtävää varten se hankitaan. Hyvä aloitustapa on tiedustella onko kenelläkään kokemuksia ao. kehittäimestä ja onko sillä tehty toimivia järjestelmiä. Tämä ei saa kuitenkaan olla tärkein valintakriteeri, sillä erilaisiin ongelmiin sopivat erilaiset kehittimet.

## 6.2 Valittavalle kehittimelle asetettavia vaatimuksia

Tehtävä, jota varten asiantuntijajärjestelmä rakennetaan, ratkaisee paljon millainen kehitin on ongelmaan sopivin. Jos kehitin hankitaan koulutus- ja harjoittelutarkoituksiin, on helppokäyttöinen ja erilaisiin ongelmatyyppeihin soveltuva kohtuuhintainen kehitin todennäköisesti paras ratkaisu. Jos aj rakennetaan kaupalliseen tarkoitukseen, jossa aj:stä saatava taloudellinen hyöty on merkittävä, niin kehittimen hinta, joka vaihtelee tuhannesta markasta muutamaan kymmeneen tuhanteen markkaan, ei varmaankaan ole tärkein valintakriteeri./18/

Ohjelmistotuki on erittäin tärkeä seikka, jota ei sovi väheksyä kehitintä valittaessa. Ohjelmistotuki ja mahdollinen kehittimen edustajan tai maahantuojaan järjestämä koulutus ovat avainasemassa, jos kehitintä halutaan käyttää mahdollisimman tehokkaasti. /18/

### 6.21 Kehittäjäliityntä

Teknisistä ominaisuuksista tärkein on aj:n kehitysvaiheessa kehittäjäliityntä, joka sisältää välineet tietämyskannan luontiin, debuggaus eli virheenetsintävälineet, välineet käyttäjäliitynnän tekemiseen ja päättelyn ohjausmahdollisuudet. Helppokäyttöisenä kehittäjäliityntä nopeuttaa sääntöjen kirjoittamista, tietämyskannan tutkimista ja ominaisuuksien muuttamista. Hyvissä kehittimissä kehittäjäliityntä on toteutettu menupohjaisena, jolloin kehittäjän ei tarvitse muistaa esim. sääntöjen tai väittämien syntaksia tai komentokielen käskyjä. /23/

### 6.22 Käyttäjäliityntä

Käyttäjäliitynnän ominaisuudet vaikuttavat lopullisen aj:n ominaisuuksiin. Hyvän käyttäjäliitynnän ominaisuuksia ovat



erilaiset kyselymahdollisuudet, kuten MIKSI ja MITEN, joilla käyttäjä pystyy tutkimaan miksi jotain tietoa tarvitaan tai miten johonkin lopputulokseen on päädytty. Myös eri tilojen talletus-mahdollisuudet ja 'Entä jos'-kyselyt monipuolistavat käyttäjän mahdollisuuksia. Päättelypuun graafinen esitys on myös tarpeellinen ominaisuus./23/

'Entä jos'-kyselyllä käyttäjä voi muuttaa monista lähtötiedoista vaikkapa vain yhtä ja katsoa miten lopputulos muuttuu tuon yhden muutoksen seurauksena.

Päättelypuun graafinen esitys on erittäin hyödyllinen järjestelmän selittäessä miten lopputulokseen on päästy. Päättelypuussa on piirretty lopputulokseen johtaneiden sääntöjen riippuvaisuussuhteet.

### 6.23 Ohjelmointikieli

Ohjelmointikieli, jolla kehitin on tehty, vaikuttaa moniin kehittimen ominaisuuksiin. Pääasiassa se vaikuttaa kehitin nopeuteen ja ulkopuolisten ohjelmien käyttöön. Nopeuteen ohjelmointikieli vaikuttaa siten, että jos se voidaan kääntää konekielelle, se on huomattavasti tulkittavaa ohjelmointikieltä nopeampi. Ulkopuolisia funktioita ja ohjelmia voidaan tehdä samalla ohjelmointikielellä kuin millä aj on toteutettu, jolloin niiden käyttö helpottuu ja rakennettava aj saadaan monipuolisemmaksi./23/

Tiivistettynä yhdeksi listaksi kehitin valintakriteerit voisivat olla seuraavat:

- 1) Ovatko kehitin tietämyksen esittämistekniikat tarpeeksi kehittyneet ja soveltuuko se ko. ongelman ratkaisuun



- 2) Onko kehittäjäliityntä hyvä eli onko kehittimellä helppo rakentaa tietämuskanta sääntöineen
- 3) Onko käyttäjäliityntä hyvä eli onko rakennettua aj:ää helppo käyttää
- 4) Onko kehittimen tuki eli ylläpito järjestetty
- 5) Onko kehitin tarpeeksi nopea ja ovatko liittynät ulkopuolisiin ohjelmiin riittävät
- 6) Onko hinta-ominaisuus suhde hyvä

Jos markkinoilla ei ole sillä hetkellä sopivaa kehitintä, eikä aj:n rakentamisella ole kiirettä, kannattaa odottaa vuoden verran, sillä uusia yhä parempia kehittämiä syntyy joka vuosi.

### 6.3 Markkinoilla olevia mikrotietokonepohjaisia asiantuntijajärjestelmäkehittämiä

Tässä käsiteltävät kehittimet toimivat IBM-PC-yhteensopivissa mikrotietokoneissa.

Kirjallisuudesta tutkittiin markkinoilla olevia mikropohjaisia kehittämiä. Markkinat muuttuvat nopeasti eivätkä lehdet artikkeleineen näytä pysyvän aina vauhdissa mukana. Monista kehittimistä oli tullut uusia versioita, joita ei kirjallisuudessa oltu arvioitu.

VP Expert, Nexpert Object, Insight 2+, Xi Plus, Personal Consultant ja Acquaint ovat PC:ssä toimivia kehittämiä, joista Nexpert Object, Acquaint ja Personal Consultant ovat kehyspohjaisia ja VP Expert, Insight 2+ ja Xi Plus ovat sääntöpohjaisia. Kehyspohjaiset järjestelmät ovat huomattavasti

tavasti sääntöpohjaisia kehittämiä kalliimpia, eikä tähän sovellukseen (murskausharjoitus) kehyspohjainen tietämyksen esittämistekniikka sovellu. Kolmesta sääntöpohjaisesta kehittäimestä valittiin VP Expert ja Xi Plus kokeiltavaksi. /16, 23, 30, 31/

### 6.31 VP-Expert

Yksinkertaisia sääntöpohjaisista kehittäimistä VP Expert on mielenkiintoinen lukuisine ominaisuuksineen ja huomattavan edullisena se on varteenotettava vaihtoehto, jos kehittäjä haluaa hankkimassa koulutustarkoituksiin. VP Expertissä on monipuoliset liityntämahdollisuudet ulkopuolisiin ohjelmiin, kuten dBase III ja Lotus 123 liitynnät. Lyhyen kokeilun jälkeen VP-Expert ei kuitenkaan osoittautunut kovin käyttäjäystävälliseksi. VP-Expertin oma tekstieditori on hyvä ja tietämyskannan ajaminen käy nopeasti. Virheilmoitukset eivät ole kovin selvittäviä. VP Expertissä on myös induktiivinen sääntögeneraattori, jolla voidaan luoda nopeasti sääntöjä esimerkiksi jonkintaulukkolaskentaohjelman taulukoista. /24, 32/

### 6.32 Xi Plus

Xi Plus on sääntöpohjainen kehittäjä, jolla on monipuoliset ominaisuudet. Päättelymekanismeina on sekä eteen- että taaksepäinpäätely. Siinä on liitynnät ulkopuolisiin ohjelmiin (dBase III ja Lotus sekä C-kielen funktiot). Xi Plus poikkeaa muista markkinoilla olevista kehittäimistä suomenkielisyydellään. Säännöt kirjoitetaan suomen kielellä, joten tietämyskannan lukeminen on erittäin helppoa, mutta joidenkin käsitteiden suomentaminen saattaa aluksi aiheuttaa hämmennystä varsinkin, kun suurin osa alan kirjallisuudesta on englanniksi. Tietämyskannan suomenkielisyys mm. vähentää dokumentointitarvetta. Xi Plussassa on hyvä kehittäjäliityntä ja erittäin hyvät mahdollisuudet



tehdä menupohjainen käyttäjäystävällinen käyttäjäliityntä. Järjestelmän oppii nopeasti ja kun myyjä tarjoaa vielä tukea, niin järjestelmä tuntui järkevältä valinnalta tähän tehtävään./33/

#### 6.4 Valitun asiantuntijajärjestelmäkehittimen ominaisuuksia

Xi Plus valittiin käytettäväksi kehittimeksi. Sen vahvoja puolia on erityisesti kehittäjäliityntä monipuolisine työkaluineen. Kehittäjäliityntä on menupohjainen, mutta käskyjä voidaan kirjoittaa myös komentoikkunaan, joka nopeuttaa kehittimen käyttöä. Myös kehittimen editori on hyvä. Editorissa on tietämyskannan kirjoittamista helpottavia ja nopeuttavia toimintoja kuten 'etsi/korvaa' , 'siirrä', 'kopioi' jne. Muita hyviä toimintoja on mm. funktionäppäimen painalluksella saatava lista tietämyskannan käsitteistä, joten kehittäjän ei tarvitse tehdä muistiinpanoja käytettävistä käsitteiden nimistä. Tietämyskanta voidaan tarkistaa esim. ristiriitojen varalta tai tutkia sääntöjen käyttämättömät niin-osat.

Käyttäjältä kysyttäviä tietoja varten voidaan tehdä erilaisia lomakkeita, jotka ovat helppotajuisia ja selkeitä. Lomakkeen koko voidaan määritellä itse ja siihen voidaan liittää uusia lomakkeita tai pinota lomakkeita päällekkäin. Myös lomakkeiden värit voidaan valita. Lomakkeisiin on mahdollista liittää myös 'Help'-tiedostot, jotka opastavat käyttäjää tarvittaessa. Tulostettavaa tai syötettävää tietoa varten lomakkeelle määritetään yksi tai useampi kenttä. Syöttökenttä määritellään joko valikko-, teksti-, numeerinen- tai Kyllä/ei-vastaus-tyyppiseksi. Tulostuskenttä voidaan määritellä numeerista tai tekstitietoa varten tai raporttialueeksi. Hyvä ominaisuus on myös se, että sama kenttä voi olla sekä tulostus- että syöttökenttä. Tämä



piirre on käyttökelpoinen tapauksissa, joissa tietyt ehdot täyttävistä vaihtoehtoista käyttäjän tulee valita yksi. /33/

Xi Plussan tietämyksen esittäminen tapahtuu sääntöjen, oletuksien ja faktojen avulla. Järjestelmässä voidaan käyttää myös muuttujia, jotka eroavat tavallisista käsitteistä siten, että ne kirjoitetaan tietämyskantaan alkamaan isolla kirjaimella. Muuttujat voivat olla yksi- tai moniarvoisia. Moniarvoiset muuttujat ovat erittäin käyttökelpoisia tapauksissa, joissa tiettyyn ryhmään kuuluu useita yksilöitä. Esimerkiksi tapaus, jossa tietyt ehdot täyttäviä murskaimia on useita voidaan kaikki pistää saman joukon alle esimerkissä olevalla säännöllä. Muuttujille voidaan määrittellä 'kehysmäisesti' erilaisia attribuutteja, liittämällä attribuutit muuttajan perään ja erottamalla ne kaksoispisteellä. Näitä kaikkia edellä mainittuja ominaisuuksia selventää seuraava esimerkki sääntö.

jos Mmurskain kuuluu murskaimiin

ja Mmurskain : kap\_yläraja < välimurskaimen kapasiteetti

ja Mmurskain : kap\_alaraja > välimurskaimen kapasiteetti

ja Mmurskain : edellä oleva max\_asetus >= välimurskaimen  
asetus

ja Mmurskain : min\_asetus <= valittu jälkimurskaimen asetukset

ja Mmurskain : max\_asetus >= valittu jälkimurskaimen asetukset

ja Mmurskain : syöttöaukko > suurin välimurskaimen  
läpäissyt raekoko

niin mahdollisesti sopiviin jälkimurskaimiin kuuluu Mmurskain

Tässä säännössä on muuttujana Mmurskain, jonka attribuutteja ovat kap\_yläraja, kap\_alaraja, edellä oleva max\_asetus, min\_asetus, max\_asetus ja syöttöaukko. Säännön niin-osassa oleva 'kuuluu'-relaatio on määritelty moniarvoiseksi, joten 'mahdollisesti sopiviin murskaimiin' voi kuulua useita murskaimia samanaikaisesti.

Relaatioita voidaan määrittää itse ja niille voidaan määrittää myös kielteiset muodot. Esimerkiksi 'kuuluu'-relaation kielteiseksi muodoksi voidaan määritellä 'ei kuulu'. Kielteistä muotoa ei kuitenkaan voi jostain syystä käyttää sääntöjen 'niin'-osissa, jolloin ne menettävät osan merkityksestään.

Xi Plussassa voidaan yhdessä sovelluksessa käyttää useita tietämyskantoja, jotka voidaan ladata toisen tietämyskannan sisältä. Tämä tapahtuu yksinkertaisesti antamalla komento 'lataa tietämyskanta'. Uusi tietämyskanta voi jatkaa päättelyä niistä tiedoista mihin edellinen tietämyskanta lopetti tai se voi aloittaa nollilta. Yksittäinen tietämyskanta voi toimia myös eräänlaisena aliohjelmana, joka vastaanottaa ja palauttaa vain haluttujen käsitteiden arvoja./33, 34/

Xi Plussan versio 3.0 on toteutettu C-kielillä ja se on huomattavasti nopeampi kuin Prologilla kirjoitettu versio 2.0. Suomenkielisessä versiossa oli kehittäimen selitysmekanismissa häiriötä. Uusi englannin kielinen versio toimi sen sijaan virheettömästi. Eräs mielenkiintoinen piirre oli se, että suomeksi kirjoitettu tietämyskanta voitiin siirtää suoraan englanninkieliseen versioon ilman välivaiheita. Sama pätee myös toisin päin. Tästä on etua, jos hankkii uuden version kehittäimestä ja jos aikaisemmat tietämyskannat on kirjoitettu suomeksi, niin ne voidaan välittömästi siirtää uudelle versiolle.

## 7. MURSKAUSHARJOITUS ASIANTUNTIJAJÄRJESTELMÄ MURMUR

### 7.1 Yleistä

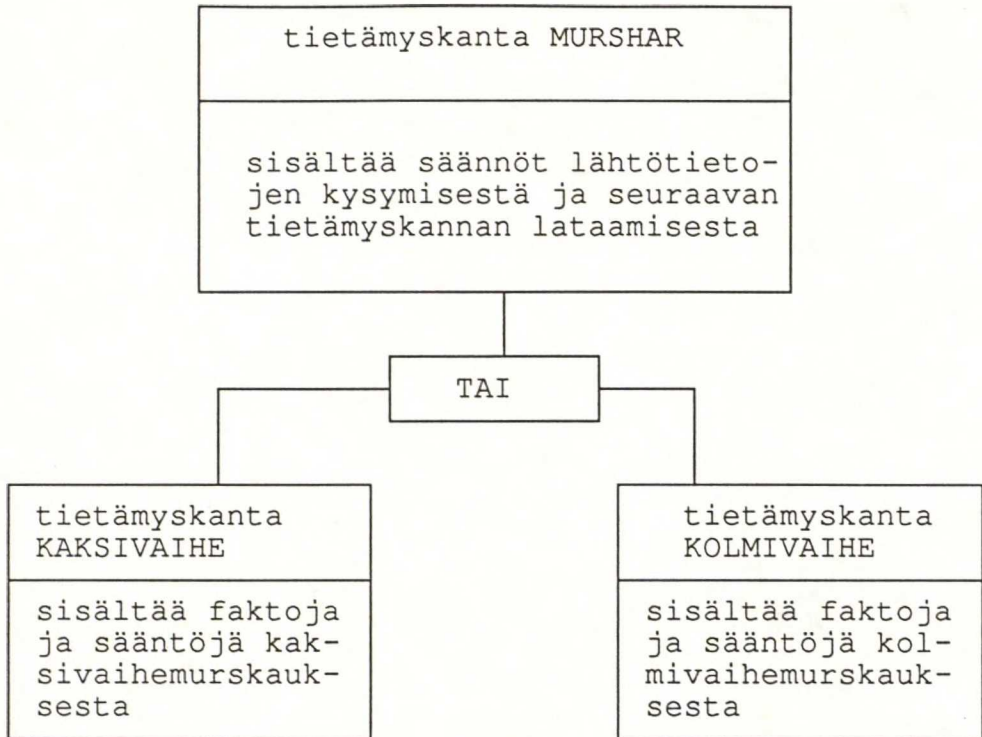
Murskauksessa käytetään yleensä useampia murskaimia (yksi-, kaksi-, kolmi- ja nelivaihemurskaus). Murskaimia kutsutaan esi-, väli- ja jälkimurskaimiksi sen mukaan missä vaiheessa niitä käytetään. Murskaimia on eri tyyppisiä ja ne voidaan jakaa rakenteen ja toimintaperiaatteen mukaan seuraaviin ryhmiin: leukamurskaimet, karamurskaimet, kartiomurskaimet, valssimurskaimet, iskumurskaimet, vasaramurskaimet ja linkomylllyt./10/

Maa- ja kalliorakennustöiden murskausharjoitus soveltuu hyvin mikropohjaisen asiantuntijajärjestelmän kehittämisen esimerkiksi. Tehtävä koostuu useista valinnoista, jotka tehdään diagrammien ja taulukoiden avulla. Tehtävä on hyvin rajattu, ristiriidaton ja matemaattiset laskut ovat niin yksinkertaisia, että ne voidaan suorittaa kehittimen omilla funktioilla. Asiantuntijajärjestelmä MURMUR on rakennettu Xi Plus-kehittimellä. Järjestelmä käyttää pääosin taaksepäinpäätelyä. Järjestelmän tietämyskannan demonit, jotka ohjaavat suoritusjärjestystä käyttävät tosin eteenpäinpäätelyä. Tämä asiantuntijajärjestelmä on tyypillinen 'teknikojärjestelmä', joka sisältää 'sanakirjatietämystä'. Sanakirjatietämyksellä tarkoitetaan kirjallisuudesta löydettävää tietämystä, kuten normeja, diagrammeja jne. MURMUR on asiantuntijajärjestelmä, jota esimerkiksi opiskelijat voivat käyttää harjoituksissa tai assistentit harjoitustehtävien tarkastamisessa. Järjestelmä nopeuttaa huomattavasti tehtävän suorittamista tai tarkastamista ja tekee valinnat ja laskut virheettömästi.



## 7.2 MURMUR

Maa- ja kalliorakennustöiden murskausharjoituksessa annetaan lähtötietoina louheen raekoko ja louheen määrä tonneissa. Tehtävänä on kiviaineksen murskaaminen halutuiksi lajitteiksi valitsemalla sopivat murskaimet. Tässä harjoituksessa tehdään yleensä kolme eri lajitetta. Järjestelmässä oletetaan, että tehtäviä lajitteita on aina kolme, esim. 0-4, 4-12 ja 12-20 mm. Tehtävät lajitteet käyttäjä valitsee MURMURin valikoista. Kun lähtöarvot on syötetty, järjestelmä aloittaa päättelyn. Järjestelmässä on kolme tietämyskanta, joista yhdellä kertaa käytetään kahta. Ensimmäisenä on käytössä perustietämyskanta, jossa on lähtötietojen syöttämisestä koskevia sääntöjä sekä säännöt siitä, mikä tietämyskanta ladataan seuraavaksi. Seuraava tietämyskanta valitaan sen mukaan, käytetäänkö kaksi- vai kolmivaihemurskausta (kuva 20). Kaksi- ja kolmivaihemurskausten tietämyskannat ovat muuten samanlaisia, mutta kaksivaihemurskauksen tietämyskannasta puuttuu välimurskainten valintaa koskevat säännöt ja faktat. MURMUR pyytää käyttäjää valitsemaan päättelyprosessin aikana murskaimien asetuksia ja iskunpituuksia. Jos sopivia murskainvaihtoehtoja on useampia, pyydetään käyttäjää valitsemaan yksi sopiva. Tarvittaessa MURMUR antaa neuvoja, jos käyttäjä ei ymmärrä suoritettavaa valintaa. Lopputuloksina saadaan eri murskausvaiheiden murskaussuhteet, murskaukseen kuluvat ajat ja syntyneiden lajitteiden määrät.



Kuva 20. MURMURin tietämyskannat

Eri murskausvaiheissa kuluvat ajat pyritään saamaan yhtä pitkiksi, jotta laitoksen kapasiteetti käytettäisiin mahdollisimman tehokkaasti. Tätä kutsutaan murskauslaitoksen tahdistamiseksi. Murskauslaitoksen tahdistus voidaan tehdä muuttamalla murskaimien asetuksia, iskunpituuksia tai valitsemalla kokonaan eri murskain, jolla on suurempi tai pienempi kapasiteetti. Nämä muutokset voidaan tehdä Xi Plussan 'Entä Jos' (What If)-toiminnolla. Tämä on erittäin tehokas piirre ja soveltuu erinomaisesti juuri tällaiseen iteratiiviseen kokeiluun, jossa ainoastaan yhtä arvoa muuttamalla voidaan kokeilla sen vaikutusta lopputulokseen.

Ensimmäiseksi valitaan esimurskain. Valinta suoritetaan yksinkertaisesti louheen suurimman raekoon mukaan, eli valitaan murskain siten, että louhe mahtuu murskaimen kitaukosta. MURMURin säännöt esimurskaimen valinnasta ovat

seuraavanlaisia

jos lähtömateriaalin raekoko  $\geq 550$

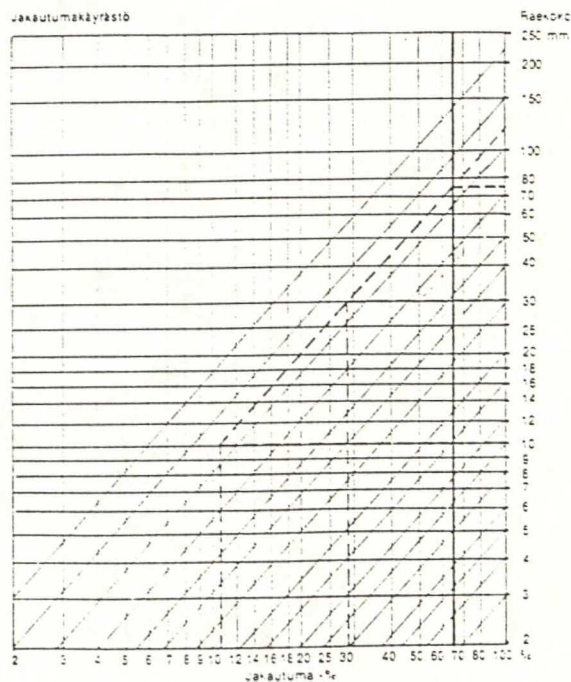
ja lähtömateriaalin raekoko  $< 700$

niin sopiviin esimurskaimiin kuuluu lokomo c100

ja sopiviin esimurskaimiin kuuluu lokomo c1 80

Jos sopivia esimurskaimia on kaksi, kuten edellä, käyttäjän tulee valita niistä toinen. Tämän jälkeen käyttäjää pyydetään valitsemaan esimurskaimen asetus. Vaihtoehdot määräytyvät valitun esimurskaimen mukaan. Esimurskaimen asetus määrää murskaimelta syntyvän murskeen raekoon jakautuman ja vaikuttaa myös murskaimen kapasiteettiin. Asetus määrää myös suurimman raekoon, joka läpäisee esimurskaimen.

Esimurskauksessa syntyneiden tuotteiden osuudet saadaan Lokomo C-sarjan kiertomurskainten jakautumakäyrästä (ns. Richardsin diagrammi, kuva 21).



Kuva 21. Jakautumakäyrästä eli Richardsin diagrammi /9/



Käyristä katsottiin eri raekokojen jakautumat eri asetuksen arvoilla ja ne kirjoitettiin Xi Plussan säännöiksi.

jos esimurskaimen asetus on 40

niin  $x_{lajite4} = 0.08$

ja  $x_{lajite6} = 0.12$

ja  $x_{lajite8} = 0.15$

ja  $x_{lajite10} = 0.18$

ja  $x_{lajite12} = 0.20$

ja  $x_{lajite16} = 0.29$

ja  $x_{lajite20} = 0.35$

ja  $x_{lajite25} = 0.43$

ja  $x_{lajite32} = 0.53$

ja  $x_{lajite55} = 0.90$

ja suurin esimurskaimen läpäissyt raekoko = 60

Jos käytetään kolmivaihemurskausta valitaan seuraavaksi sopiva välimurskain. Kaksivaihemurskauksessa siirrytään suoraan jälkimurskaimen valintaan. Välimurskainta valittaessa vaatimuksina ovat edellä olevan murskaimen maksimi asetus sekä suurin esimurskaimen läpäissyt raekoko. Järjestelmän karamurskainten tiedot ovat tietämyskannassa faktoina. Järjestelmä valitsee kaikki vaatimukset täyttävät murskaimet "mahdollisiin välimurskaimiin kuuluu"-joukkoon, josta käyttäjä voi valita yhden sopivan. Kun sopiva välimurskain on valittu, järjestelmä pyytää käyttäjää valitsemaan välimurskaimessa käytettävän iskunpituuden ja asetuksen. Iskunpituus- ja asetusvaihtoehdot ovat sidottuja valittuun murskaimeen. Toisin sanoen jos ko. murskaimessa on iskuliikevaihtoehtoina 12, 16 ja 20 mm, niin vain nämä ilmestyvät valikkoon, josta käyttäjä voi valita yhden (kuva 22).

Tietämyskanta: kolmivaihe

|  |   |
|--|---|
| <p><b>välimurskaimen valinta</b></p> <p>Sopivia välimurskaimia ovat</p> <div style="border: 1px solid black; padding: 2px; margin: 5px 0;">g 3210</div> <div style="border: 1px solid black; padding: 2px; margin: 5px 0;">g 4214</div> <div style="border: 1px solid black; height: 100px; width: 100%;"></div> <p style="text-align: center;">valitse yksi</p> | <p><b>välimurskaimen iskunpituus</b></p> <p>Valitun välimurskaimen iskunpituudet ovat :</p> <div style="display: flex; align-items: center; margin-top: 10px;"> <div style="border: 1px solid black; padding: 2px; margin-right: 10px;">16</div> <div>valitse yksi</div> </div> <div style="border: 1px solid black; padding: 2px; margin: 5px 0;">22</div> <div style="border: 1px solid black; padding: 2px; margin: 5px 0;">26</div> |
|--|---|

**välimurskaimen asetus**

35

40

45

50

55

Valitse myös välimurskaimen asetus

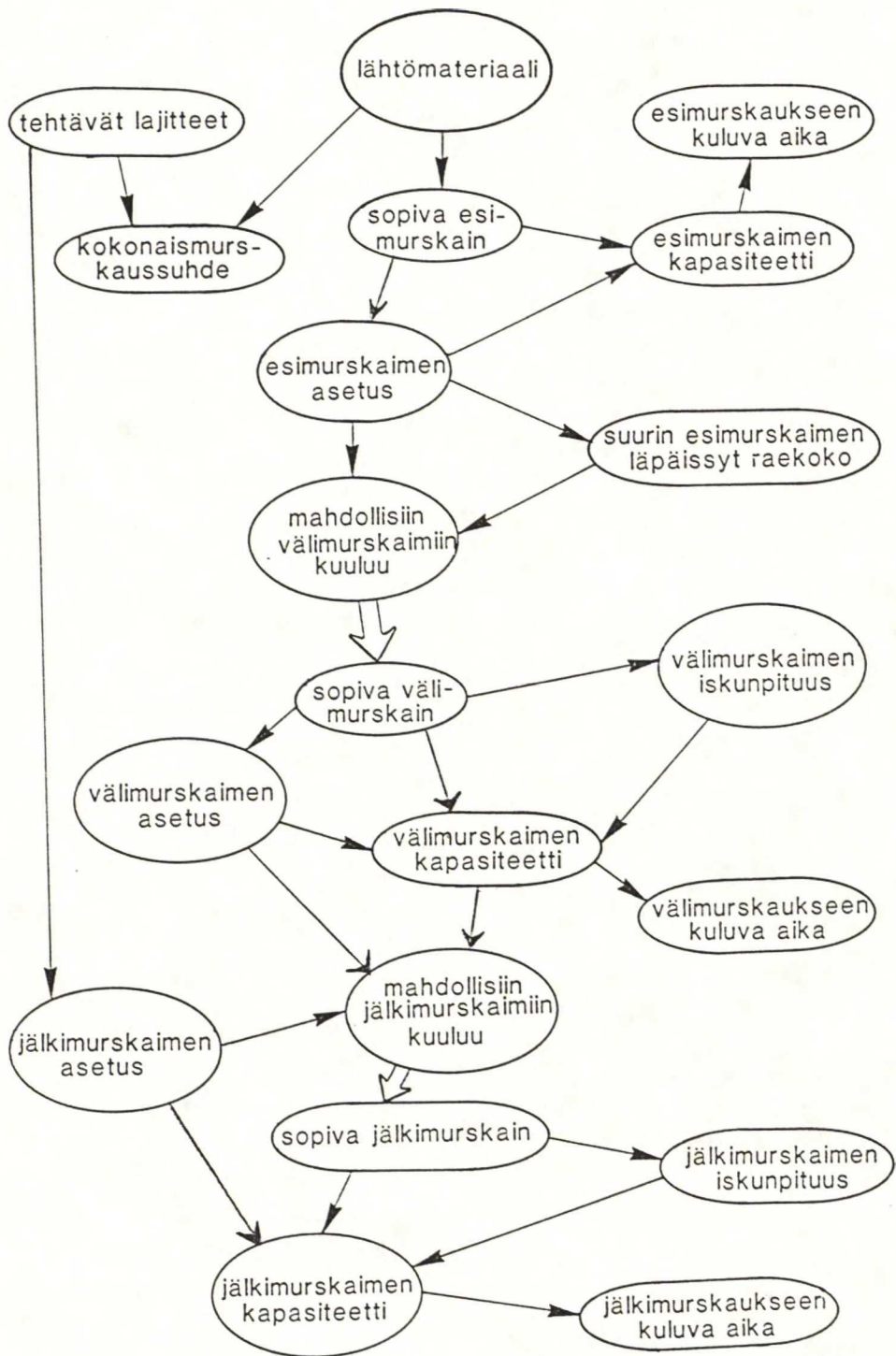
Paina F1  
lisätietoa asetuksen  
vaikutuksesta

uuden

kuva 22. MURMURin lomake 'välimurskaimen ja sen iskuliikkeen sekä asetuksen valinta'

Jälkimurskain valitaan samoista karamurskaimista kuin välimurskainkin. Tehtävien lajitteiden suurin raekoko on jo etukäteen määrännyt jälkimurskaimen asetuksen. Tämä on otettava huomioon mahdollisia jälkimurskaimia valittaessa, että tämä asetus kuuluu valittavan murskaimen asetusvaihtoehtoihin. Kaikki sopivat murskaimet esitetään ja käyttäjä valitsee niistä yhden. Valittuaan sopivan jälkimurskaimen käyttäjän pitää valita vielä murskaimen iskuliike.

Kun kaikki murskaimet on valittu ja niiden asetukset ja iskuliikkeet valittu MURMUR laskee eri murskaimilta syntyneiden lajitteiden määrät, murskainten kapasiteetit ja murskaukseen kuluvat ajat. Koko ongelma on esitetty yksinkertaistettuna kaaviona kuvassa 23.



Kuva 23. MURMUR yksinkertaistettuna verkkorakenteena.



## 8. JÄRJESTELMÄN ARVIOINTI

### 8.1 Yleistä

MURMURin tietämuskanta on rakennettu Maa- ja kalliorakennustöiden murskausharjoituksessa annettavien tietojen pohjalta. Harjoituksessa on käsitelty ainoastaan Lokomon murskaimia, joten järjestelmä soveltuu tässä laajuudessa ainoastaan opiskelukäyttöön.

Esimurskaimina on käytettävissä Lokomon C-sarjan murskaimia, joista valitaan sopivat murskaimet louheen suurimman raekoon perusteella.

Esimurskaimen kapasiteetti on arvioitu taulukoiden arvojen keskiarvoina, esim 240-300 m<sup>3</sup>/h rajoista saadaan kapasiteetiksi 270 m<sup>3</sup>/h. Käytännössä kapasiteettiin vaikuttaa mm. materiaalin laatu, jakautuma ja syöttötapa, mutta tässä järjestelmässä näiden vaikutusta ei ole otettu huomioon. MURMURin säännöt esimurskaimen kapasiteetista ovat seuraavanlaisia :

jos sopiva esimurskain on lokomo c160

niin esimurskaimen kapasiteetti =  $2.1 \cdot \text{esimurskaimen asetus} - 30$

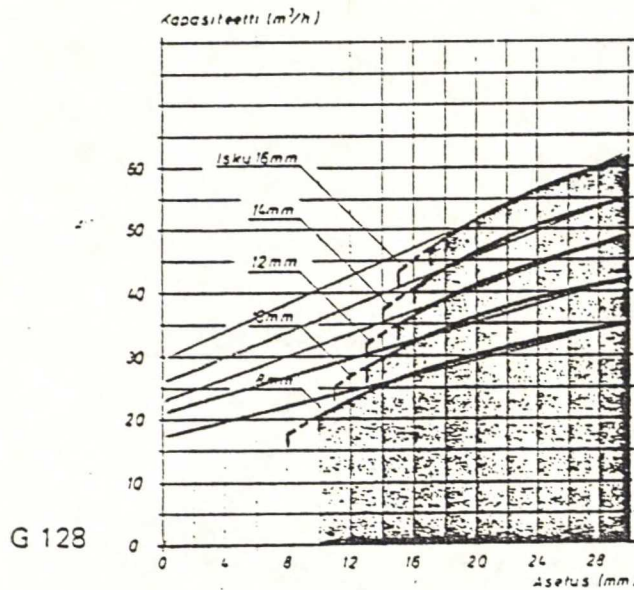
Väli- ja jälkimurskainten kapasiteettikäyristä on muodostettu funktiot, jotka on kirjoitettu järjestelmän sääntöihin. Kapasiteetti riippuu murskaimen asetuksesta ja murskaimen iskuliikkeen pituudesta. Kapasiteettia koskevat säännöt ovat seuraavanlaisia :

jos sopiva välimurskain on g 1010

ja valittu välimurskaimen iskunpituus on 22

niin välimurskaimen kapasiteetti =  $0.83 \cdot \text{välimurskaimen asetus} + 64$

Kun kapasiteettikäyriä muokattiin funktioiksi, niitä yksinkertaistettiin hieman, piirtämällä suorat mahdollisimman tarkasti käyrien päälle. Näiden suorien avulla muodostettiin matemaattiset yhtälöt, joiden paikkansa pitävyys tarkistettiin. Virhettä syntyy lähinnä käyrien alkuosalla, ks kuva 24, mutta se jää merkityksettömän pieneksi.



kuva 24. Murskaimien kapasiteetin riippuvaisuus iskuliikkeestä ja asetuksesta/9/

Väli- ja jälkimurskainten jakaumakäyristä kirjoitettiin säännöt samaan tapaan kuin esimurskaimen jakaumakäyristä. Välimurskaimen raekoon jakautuma riippuu käyttäjän valitsemasta asetuksesta. Jälkimurskaimella asetus määräytyy suurimman raekoon mukaan ja raekoon jakautuma riippuu tästä asetuksesta.

Testitapauksista MURMUR suoriutui hyvin. Käsien lasketut murskausharjoitukset ajettiin MURMURilla ja jos eroavuuksia tuli, olivat virheet aina käsien tehdyissä harjoituksissa.

## 8.2 Kehittäjäliityntä

Jos käyttäjä haluaa täydentää MURMURin tietämyskantaa lisäämällä uusia sääntöjä se käy helposti hyvän editorin avulla. Myös järjestelmän käyttäjäliityntää voidaan muokata. Xi Plussassa on erittäin hyvät työkalut selkeän käyttäjäliitynnän tekemiseen, joita on käsitelty tarkemmin luvussa 6.4. Uusia sääntöjä lisättäessä on hyvä käydä MURMURin nykyinen tietämyskanta läpi ja tarkistaa, ettei uusista säännöistä aiheudu ristiriitoja.

## 8.3 Käyttäjäliityntä

MURMURin käyttäjäliityntä on rakennettu menupohjaiseksi. Käyttäjälle tulee näkyviin lomakkeita, joissa on vain siihen asiaan liittyvää tietoa. Näin saadaan käyttäjältä kysyttävä ja käyttäjälle ilmoitettava tieto jaettua sopiviin paloihin (kuva 25).

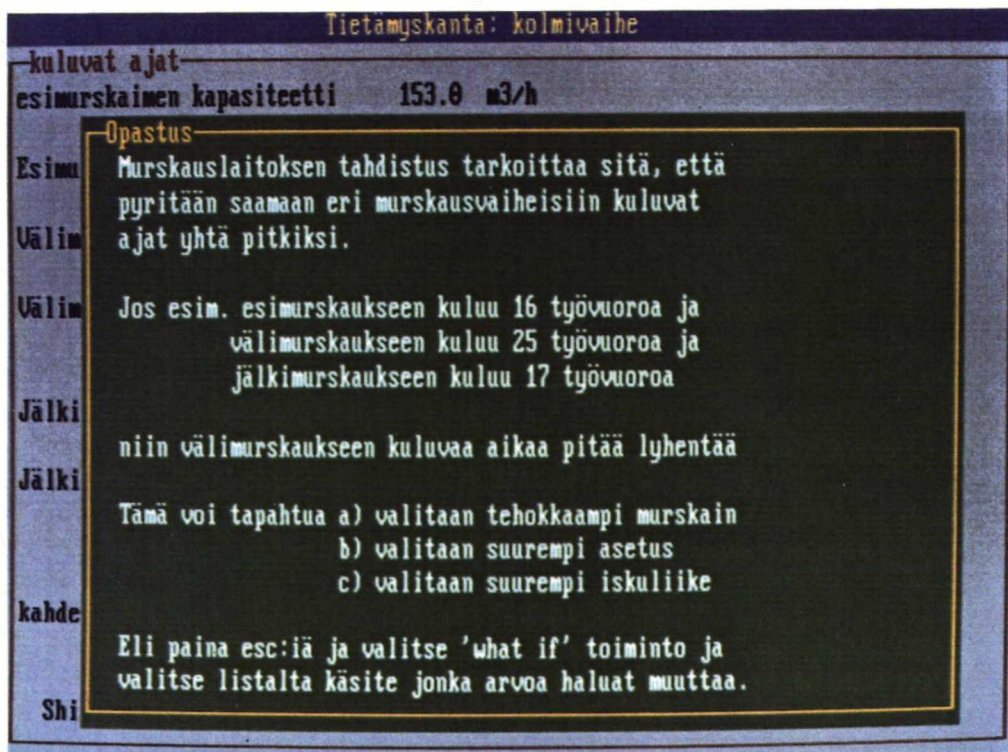
| Tietämyskanta: murhar  |   |  |
|--|---|--|
| tehtävä lajite 1   |   |  |
| <b>Valitse ensimmäinen lajite</b>  | <b>Valitse toinen lajite</b>  | <b>Valitse kolmas lajite</b>   |
| <ul style="list-style-type: none"> <li>0...4 mm</li> <li>0...6 mm</li> <li>0...8 mm</li> <li>0...10 mm</li> <li>0...12 mm</li> </ul> | <ul style="list-style-type: none"> <li>8...12 mm</li> <li>8...16 mm</li> <li>8...20 mm</li> </ul> | <ul style="list-style-type: none"> <li>12...20 mm</li> <li>12...25 mm</li> <li>12...32 mm</li> <li>12...55 mm</li> </ul> |

kuva 25. MURMURin lomake 'tehtävät lajitteet'

Käyttäjä voi tarvittaessa pyytää apua, joka tapahtuu funktionäppäintä painamalla. Jos painetaan funktionäppäintä F1 niin kuvaruudulle ilmestyy käsitteeseen liittyvä 'apu-



raportti' tai jos painetaan shift F1 niin ilmestyy lomakkeeseen liittyvä 'apu-raportti', joka opastaa käyttäjää (kuva 26).



kuva 26. Käyttäjän opastus 'Help'-toiminnolla

Käyttäjä voi kesken konsultoinnin kysyä myös miksi jotain tietoa tarvitaan tai kysyä miten johonkin tulokseen on päästy. Myös sen hetkiset tiedot saadaan kuvaruudulle. Käyttäjä voi myös kirjoittaa komentoikkunaan jonkun käsitteen nimen ja kysymysmerkin, jolloin järjestelmää antaa kyseisen käsitteen arvon.

#### 8.4 Muuteltavuus

MURMURin tietämyskanta on helppo muuttaa. Uusia sääntöjä lisättäessä on huolehdittava siitä, että ne eivät aiheuta ristiriitaa olemassaolevien sääntöjen kanssa. Tietämyskannan koon kasvaessa sen hallitseminen vaikeutuu. Kehittäjän

on helpompi työskennellä, jos mielessä ei tarvitse pitää koko ongelman käsitteitä vaan voidaan keskittyä johonkin osa-alueeseen kerrallaan. Jos järjestelmästä kasvaa kovin suuri, kannattaa tietämyskanta jakaa erillisiin tietämyskantoihin. Xi Plussassa voidaan tietämyskannan sisältä kutsua toista tietämyskanta ja pitää muistissa siihen asti päätellyt tiedot, joiden avulla päättelyä jatketaan. MURMURin tietämyskanta kasvoi lopulta niin suureksi, että tietämyskannan jako pienempiin osiin tuli harkittavaksi. Tietämyskanta jaettiin kahteen osaan, jolloin tuli ratkaistua helposti myös se ongelma, kuinka menetellään jos kyseessä onkin kaksivaihemurskaus. MURMURissa on molempia tapauksia varten oma tietämyskanta, joka ladataan automaattisesti.

## 8.5 Puutteet

MURMUR olettaa, että murskauksessa tehtäviä lajikkeita on aina kolme. Tämän muuttaminen sellaiseksi, että käyttäjä saisi valita kuinka monta lajiketta tehdään kävisi helpommin perinteisellä ohjelmointikielellä, kuin sääntöihin perustuvalla aj-kehittimellä. Myös tehtävien lajikkeiden vaihtoehdot ovat rajatut. MURMURissa olevat lajikevaihtoehdot ovat yleisimmin murskausharjoituksissa esiintyvät. Murskauslaitoksen tahdistus tehdään käyttäjän toimesta muuttamalla 'Entä jos'-toiminnon avulla murskaimen arvoja. Perinteisellä atk-ohjelmalla olisi ollut helppo toteuttaa silmukka, joka muuttaa arvoja niin kauan, kunnes haluttu tarkkuus on saavutettu. Myös järjestelmän 'Miksi' ja 'Miten' toiminnot vaatisivat parannuksia. Koska käyttäjäliityntä on tehty mahdollisimman ystävälliseksi ja selkeäksi käyttämällä demoneita, jotka ohjaavat missä järjestyksessä käyttäjälle esitetään kysymyslomakkeita.

kun tehtävä murskausharjoituksen suoritus  
niin kirjoita tiedostosta alkutext.rpt  
ja suorita lomake alkutiedot  
ja suorita lomake tehtävä lajite 1 .....

Siitä aiheutuu seuraavanlaisia haittoja. Kun eri tietoja  
kysytään lomakkeiden avulla, Xi Plus vastaa kysyttäessä  
Miksi jotain tietoa tarvitaan : käytetään lomaketta 'Alku-  
tiedot' koska : suorita lomake 'Alkutiedot'  
perustuen :

kun tehtävä murskausharjoituksen suoritus  
niin kirjoita tiedostosta alkutext.rpt  
ja suorita lomake 'Alkutiedot' ...

koska eteenpäinpääteltiin käyttäen tehtävä murskaushar-  
joituksen suoritus, koska tehtävänä oli murskausharjoituksen  
suoritus.



## 9. TULEVAISUUDEN NÄKYMÄÄ

### 9.1 Yleistä

Viime vuosien tietämystekniikan nopea kehitys käytännön soveltamisen asteelle ja sen myötä ilmestyneet uudet asiantuntijajärjestelmien rakentamisvälineet ovat vain esimakua tietämystekniikan tarjoamista mahdollisuuksista. Mikrotietokoneille tehdyt kehittimet ovat tuoneet asiantuntijajärjestelmät kaikkien ulottuville ilman suuria investointeja. Sekä laitteistojen että ohjelmistojen osalta odotetaan edelleen nopeimman kehityksen tapahtuvan nimenomaan mikrotietokoneiden puolella. Ohjelmointikielissä näkyy suuntaus siirtyä käyttämään perinteisiä ohjelmointikieliä, kuten C:tä ja jopa Fortrania, asiantuntijajärjestelmäkehittimien ohjelmointikielinä, sillä niiden siirrettävyys ja yhteensopivuus eri laitteistojen kesken on erittäin hyvä. Niillä rakennetut kehittimet ovat myös huomattavasti nopeampia kuin korkeampitasoisilla ohjelmointikielillä rakennetut kehittimet. Uudempien ohjelmointikielien, kuten Lispin ja Prologin, suosion ei odoteta lähivuosina merkittävästi kasvavan arkipäivän sovelluksien rakentajien keskuudessa. Tämä johtuu osittain Lispin ja Prologin puutteellisuudesta standardoinnista. Lispin ja Prologin merkitystä ei kuitenkaan sovi aliarvioida korkean teknillisen tietä- taidon hyväksikäytössä. /1, 19/

### 9.2 Tietämystekniikan kehittyminen

Tietämystekniikan nopea kehitys sovelluksineen on luonut lisäintoa ja uusia tarpeita kehittää kaupallisia asiantuntijajärjestelmien rakentamisvälineitä. Tulevaisuudessa on odotettavissa seuraavan viiden tietämystekniikan osa-alueen kohdalta kehitystä :/19/

- tietämyksen esittäminen
- tietämyksen hankinta
- asiantuntijajärjestelmäkehittimet
- asiantuntijajärjestelmien suunnittelu
- asiantuntijajärjestelmien ohjelmointi

Tietämyksen esittäminen on avainasemassa kehitettäessä parempia asiantuntijajärjestelmiä. Olemassa olevia esitystekniikoita kehitetään ja parannetaan edelleen. Esimerkiksi sääntöpohjaisille järjestelmille pyritään saamaan uusien tietojen päättely vanhojen tietojen pohjalta. /19/

Nykyiset tietämyksen esittämistekniikat ovat rajoittuneita. Lisämahdollisuuksia saadaan erilaisten tietämyksen esittämistekniikoiden yhdistämisellä ja kehittämällä uusia tietämyksen esittämistekniikoita. Kaksi odotettavissa olevaa uutta esittämistekniikkaa ovat "model-based knowledge representation scheme" eli malli-pohjainen tietämyksen esittämistekniikka sekä "script knowledge representation scheme". Mallipohjaista tietämyksen esittämistekniikkaa kutsutaan myös analogiseksi tietämyksen esittämistekniikaksi, ks luku 2.

Tietämyksen hankinta on tärkeä tutkimusalue, jota tutkitaan sekä yliopistoissa että teollisuudessa. Asiantuntijajärjestelmien rakentamisprojekteissa tietämyksen hankinta muodostaa merkittävän osan projektin kustannuksista, varsinkin jos rakentamisessa käytetään aj-kehittintä. Jos tietämystä pystytään hankkimaan tehokkaammin ja edullisemmin pystytään koko projekti toteuttamaan halvemmallalla. Tietämyksen hankintaan käytettäviä varoja voidaan säästää seuraavin edellytyksin :/19/

- vähennetään tietämyksen hankintaan kuluva-a aikaa
- vähennetään työntekijöihin kohdistuvia pätevyysvaatimuksia
- pyritään automatisoimaan osia tietämyksen hankinnasta
- mahdollistetaan asiantuntijoiden ja käyttäjien itse lisätä tietämystä järjestelmään
- myydään tai ostetaan 'pakattua tietämystä' (packaged knowledge) tai pysyvää tietämystä (static knowledge) esimerkiksi sanakirja-tietämystä tarvittaviin sovelluksiin.

Automaattiset tietämyksen hankintatyökalut tulevat vähentämään tietämyksen hankintaprosessiin kuluva-a varoja sekä aikaa. Uusilla työkaluilla työntekijät pystyvät hankkimaan tehokkaasti tietämystä ilman pitkää koulutusta ja kokemusta.

Itse-oppivilta järjestelmiltä odotetaan tulevaisuudessa paljon. Tämän hetkiset 'itse-oppivat' työkalut ovat induktioon perustuvia, jotka muodostavat aiempien tapausten pohjalta sääntöjä. Nykyiset induktiotyökalut ovat kuitenkin niin kehittymättömiä, ettei todellisesta 'itse-oppimisesta' vielä ole kysymys. Uusilla induktioon perustuvilla 'sääntögeneraattoreilla' on löydetty suurista tietomääristä kuitenkin sellaisia riippuvaisuussuhteita, joita ihmisasiantuntijat eivät olleet huomanneet. Tämä on erittäin potentiaalinen käyttökohde esimerkiksi tutkijoille. Toinen itse-oppimisen tekniikka saattaa tulevaisuudessa olla tekstiä lukevat skannerit, jotka pystyvät löytämään oleellisen asian tekstistä ja siirtämään sen järjestelmän tietämyskantaan.

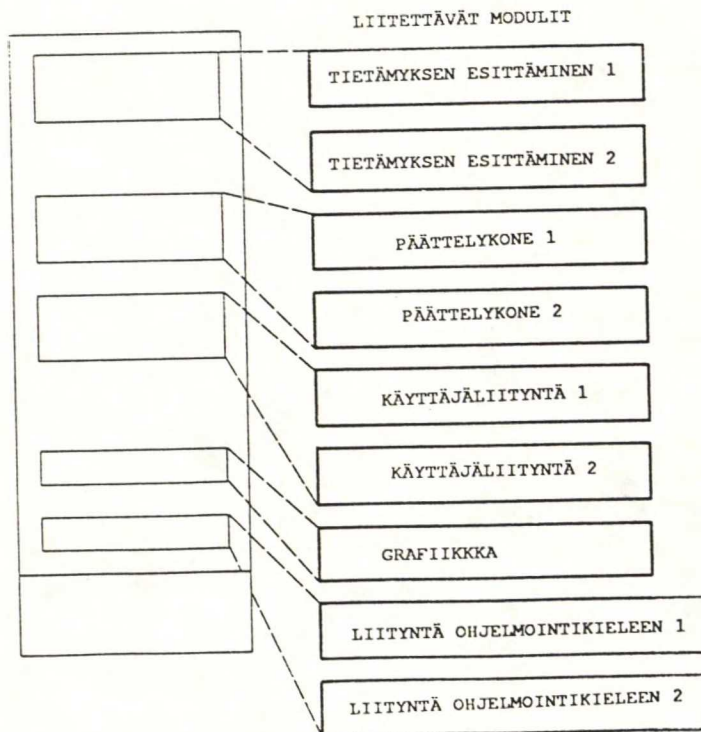
/1, 19/

Myös uusia itse-oppimisen menetelmiä tutkitaan. Yksi mielenkiinnon kohteista on dynaaminen tietämyskannan laajentaminen ja muokkaaminen. Siinä itse-oppimiskyky on integroitu toimivan asiantuntijanjärjestelmän käyttäjälii-



tyntään. Itse-oppimismoduli seuraa käyttäjän ja asiantuntijajärjestelmän välistä istuntoa. Moduli tallentaa kaikki käyttäjän vastaukset tietokantaan ja muodostaa näistä induktiolla uusia sääntöjä. / 19 /

Nykyiset asiantuntijajärjestelmäkehittimet nopeuttavat asiantuntijajärjestelmien rakentamista huomattavasti, jos verrataan asiantuntijajärjestelmän rakentamista pelkän ohjelmointikielen avulla. Tulevaisuuden kehittimet nopeuttavat järjestelmien rakentamista vielä huomattavasti. Tulevaisuuden kehittimien uskotaan koostuvan moduleista (kuva 27). Kehittimen ydin on perusmoduli, johon liittyvät erilliset toimintomodulit. Toimintomoduleina voi olla vaikkapa kaksi erillistä päättelykonetta, toinen eteenpäinpäättelyä varten ja toinen taaksepäinpäättelyä varten. Tällaisen moduleista koostuvan kehittimen etuina ovat useiden toimintomodulien käyttömahdollisuus samaan aikaan.



Kuva 27. Tulevaisuuden asiantuntijajärjestelmäkehitin /19/

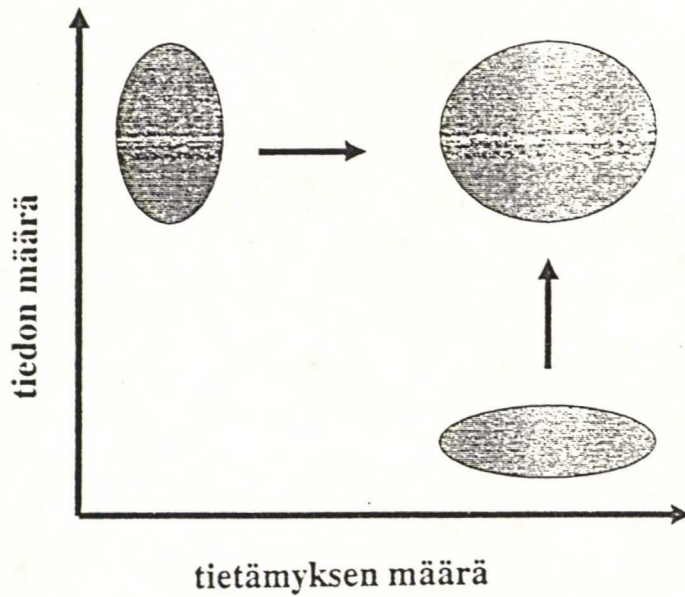
Nykyään asiantuntijajärjestelmän rakentamisprojektin suunnittelu sisältää perinpohjaisia alkuselvityksiä, jotta ongelmaan sopiva lähestymistapa löydetään. Tulevaisuudessa tähänkin on odotettavissa uusia työkaluja. Uusia työkaluja asiantuntijajärjestelmien rakentamisprojekteihin odotetaan tulevan seuraavien ongelmien ratkaisuun:/ 19/

- ongelman piirteiden määrittelyyn
- varmistamaan, että ongelma-alueen kaikki tärkeät käsitteet on huomioitu
- määrittämään mitkä tietämyksen esittämistekniikat soveltuvat kyseiseen ongelmaan parhaiten
- määrittämään mikä päättelymekanismi soveltuu parhaiten
- määrittämään käyttäjän tarpeet
- valitsemaan sopiva käyttäjäliityntätapa
- valitsemaan sopivat rakentamisapuvälineet (kehittimet), jotka soveltuvat ongelman asettamiin vaatimuksiin
- määrittämään millaisia muita ohjelmia pitää tehdä, että asiantuntijajärjestelmä saadaan toimimaan

Nykypäivänä asiantuntijajärjestelmien ohjelmointi on työläs käsin tehtävä työ. Automaattisesta ohjelmoinnista on odotettavissa apuvälineitä tähänkin ongelmaan, niitä ovat mm. :/19/

- automaattiset ohjelmointijärjestelmät
- asiantuntijajärjestelmien ohjelmointia auttavat järjestelmät
- CAM-tapaiset ohjelmointityökalut
- graafiikan ja ikonien (icon) käyttö

Tulevaisuudessa asiantuntijajärjestelmien ja perinteisten atk-ohjelmien odotetaan yhdentyvän niin, että 'älykkäillä' järjestelmillä pystytään käsittelemään suuriakin tietomääriä (kuva 28.), vrt luku 1. kuva 1. /29/



Kuva 28. Asiantuntijajärjestelmien ja perinteisten atk-ohjelmien yhdentyminen. /29/

### 9.3 Tietekniikan sovelluksien lisääntyminen

Mikrotietokonepohjaisten tietekniikan sovelluksien määrä on kasvanut koko ajan. Yhdysvalloissa on kehitetty lukuisia erilaisia laskentaan, hallintoon ja suunnitteluun liittyviä mikrotietokoneohjelmia tieinsinöörien käyttöön. Suomessa on odotettavissa voimakasta kasvua tietekniikan atk-sovelluksissa, niin perinteisten atk-sovelluksien kuin tietämystekniikan sovelluksien muodossa. Asiantuntijajärjestelmäkehittäjät ovat jo niin kehittyneitä, että niiden avulla voidaan rakentaa rajattujen ongelmien ratkaisua helpottavia järjestelmiä suhteellisen helposti. Lähivuosina odotettavissa oleva tietämystekniikan käytännön sovelluksien lisääntyminen tuo asiantuntijajärjestelmät ajankohtaiseksi jokaiselle organisaatiolle, joka haluaa pyrkiä tehokkaampaan ja korkeatasoisempaan työskentelyyn. Asiantuntijajärjestelmien avulla saadaan organisaation hallussa oleva tieto-taito levitettyä useiden henkilöiden käyttöön sekä tallennettua, jos asiantuntija on poistumassa organisaatiosta. Suomessa tietokone-



avusteinen tiensuunnittelu on jo jokaisen insinööritoimiston käytössä. Yksi potentiaallinen käyttökohde asiantuntijajärjestelmille olisi sulautettujen asiantuntijajärjestelmien (embedded expert systems) liittäminen näihin jo käytössä oleviin cad-järjestelmiin. Sulautettu asiantuntijajärjestelmä toimisi 'näkymättömänä' osana cad-järjestelmää ja valitsisi esim. normien perusteella käytettävien pyöristyskaarien enimmäis- ja vähimmäisarvot. Jälkeenpäin järjestelmä voisi perustella miksi kyseinen pyöristyskaaren arvo valittiin. Myös käyttäjäliittynän parantamiseen näissä cad-järjestelmissä voitaisiin käyttää asiantuntijajärjestelmää 'front-endinä'. Tällainen 'front-end' toimisi välikappaleena atk:hon tottumattoman käyttäjän ja cad-järjestelmän monimutkaisen käyttäjäliittynän välillä, auttaen 'älykkäästi' kokematon käyttäjää.

Tielainsäädännön eri osien siirtäminen asiantuntijajärjestelmiin tuntuisi kannattavalta hankkeelta. Lainsäädäntö ei tunnetusti ole insinöörien lempialueita, eivätkä siihen ole kovin monet insinöörit perehtyneet. Kuitenkin projektien aikana törmätään usein juridisiin ongelmiin, joiden selvittely saattaa olla hankalaa ja kallista. Monet ongelmat olisi vältetty, jos projektin alussa olisi nähty vaivaa näiden lakikiemuroiden selvittämiseksi.

## 10. PÄÄTELMÄT

Nykyisillä asiantuntijajärjestelmäkehittimillä pystytään rakentamaan toimivia järjestelmiä suhteellisen helposti. Ystävällinen käyttäjäliityntä ja helppolukuinen tietämyskanta ovat tyypillisiä piirteitä asiantuntijajärjestelmälle, joka on rakennettu nykyaikaisella kehittimellä. Tietämyskannan helppolukuisuus, kuten tässä työssä tehdyn MURMUR asiantuntijajärjestelmän suomenkielinen tietämyskanta, on ehdoton edellytys asiantuntijajärjestelmän helpolle ylläpidolle. Tietämystekniikkaa pidetään tekoälytutkimuksen osalueena, mutta järjestelmien todellisesta älykkyydestä ei voida vielä puhua. Asiantuntijajärjestelmien vaikutusta tulevaisuuden tietotekniikkaan ei pidä kuitenkaan aliarvioida, sillä tämän hetkisine puutteineenkin ne ovat erittäin käyttökelpoisia rajattujen ongelma-alueiden ratkaisuisissa. Asiantuntijajärjestelmät eivät korvaa oikeita asiantuntijoita, vaan ne ovat apuvälineitä, joilla asiantuntijan tieto-taito saadaan tehokkaaseen käyttöön.

Ryhdyttäessä rakentamaan ensimmäistä asiantuntijajärjestelmää on hyvä lähteä varovaisesti liikkeelle. Pieniltäkin näyttävät ongelmat sisältävät usein paljon enemmän pieniä 'aliongelmia' ja näiden käsitteitä, kuin ennalta osataan arvella. Jos ensimmäistä asiantuntijaprojektia käynnistettäessä ollaan liian kunnianhimoisia, saattaa valittu ongelma olla liian mutkikas ja projektin epäonnistuessa menetetään usko tietämystekniikan käyttökelpoisuuteen. Pienehköillä ongelmilla on hyvä opetella asiantuntijajärjestelmien rakentamiseen liittyviä tekniikoita, kuten tietämyksen hankintaa ja eri tietämyksen esittämistekniikoiden vertailua. Asiantuntijajärjestelmiä voidaan testata keskeneräisinäkin ja tietämyskantaa voidaan täydentää myöhemmin. Rakennetun asiantuntijajärjestelmän testaamiseen tulee kiinnittää tarpeeksi huomiota. Jos käyttäjä huomaa järjestelmän toimivan virheellisesti, hän menettää luottamuksensa



siihen eikä todennäköisesti halua enää käyttää sitä. Käyttäjät olisi hyvä saada mahdollisimman aikaisessa vaiheessa aj:n rakentamisprojektiin mukaan, jolloin heidän mielenkiintonsa saataisiin heräämään ja he ymmärtäisivät, mistä oikein on kysymys.

Valittavalle kehittimelle on asetettava vaatimuksia sen mukaan, millaiseen ongelmaan sitä käytetään. Tässä työssä tehty työkalun valinta perustui suppeaan vertailuun eikä sitä voida pitää kaiken kattavana yleisvertailuna. Nykyisten kehittimien lyhyt elinkaari vaikeuttaa kehittimen valintaa huomattavasti. Turvallinen valinta on sellainen kehitin, jolle myyjä tarjoaa käyttäjätukea ja ylläpitoa ja jolla on rakennettu toimivia asiantuntijajärjestelmiä.

Rakennettu murskausharjoitus asiantuntijajärjestelmä MURMUR todettiin käyttökelpoiseksi murskausharjoituksen suorittamiseen ja käsin tehtyjen harjoitusten tarkastamiseen. Järjestelmän tietämyskanta sisältää tiedot eri murskaimista sekä kaavat näiden kapasiteettien ja eri murskausvaiheisiin kuluvan ajan laskemiseen. Tietämyskannan hallinta tuli sitä vaikeammaksi mitä suuremmaksi tietämyskanta kasvoi. Käytetyssä kehittämissä, Xi Plussassa, oli mahdollisuus jakaa tietämyskanta erillisiksi osiksi, joka tässä tapauksessa tehtiinkin. Tämä on erittäin käyttökelpoinen ominaisuus tehtäviin, joissa ongelma voidaan jakaa ali-ongelmiksi.

Tietekniikassa on useita ongelmia, joita ei voida ratkaista perinteisen atk-tekniikan keinoilla. Monet tällaisista ongelmista on ratkaistavissa tietämystekniikan menetelmillä. Vaikka tämän hetkisisissä asiantuntijajärjestelmissä ja niiden rakentamiseen tarkoitetuissa apuvälineissä on vielä paljon parantamisen varaa, on niiden tuomat mahdollisuudet otettava huomioon kehitettäessä tietekniikan atk-sovelluksia.



Yritykset ja organisaatiot, jotka aloittavat aikaisessa vaiheessa tietämystekniikan soveltamisen käytännön ongelmiin, ovat huomattavassa etulyöntiasemassa niihin organisaatioihin nähden, jotka odottavat, kunnes aj:t ovat tarpeeksi kehittyneitä. Tekniikka kehittyy nopeasti, ja ne jotka ovat opetelleet alkeet, ovat valmiita vastaanottamaan uutta teknologiaa.

## 11. YHTEENVETO

Asiantuntijajärjestelmien kehitys on jo niin pitkällä, että kaupallisia sovelluksia on olemassa. Asiantuntijajärjestelmien rakentamisvälineiden laatu on parantunut ja erilaisia kehittämiä on saatavissa aina mikrotietokoneista mainframe-koneisiin. Korkeatasoisen tieto-taidon levittäminen organisaation sisällä onnistuu suppeiden ongelma-alueiden, kuten tievalaisimien valinta, osalta asiantuntijajärjestelmien avulla. Näin saadaan organisaation hallussa oleva tieto-taito kaikkien työntekijöiden käyttöön, joten työn tuottavuus nousee ja samalla käyttäjät oppivat ongelmien taustalla olevasta tietämyksestä ja asiantuntijat voidaan vapauttaa rutiinitehtävistä.

Asiantuntijajärjestelmien rakentamista ovat tutkineet sekä yliopistot että kaupallisesti suuntautuneet yritykset. Asiantuntijajärjestelmien rakentamiseen liittyy kiinteästi tietämyksen hankinta, jota on myös tutkittu runsaasti. Erilaisia tietämyksen hankintamenetelmiä on kehitetty ja siihen liittyviä ohjelmistoja on jo saatavilla. Suurin osa näistä ohjelmistoista perustuu induktion käyttöön. Tietämystekniikka kehittyy ja tulee yhä enemmän mukaan perinteisen atk-tekniikan rinnalle. Tulevaisuudessa näiden kahden tekniikan odotetaan yhdentyvän niin, että suuriakin tietomääriä voidaan käsitellä 'älykkäästi'. Nykyisissä mikropohjaisissa kehittimissä olevat liityntämahdollisuudet ulkopuolisiin ohjelmiin kuvaavat osaltaan sitä tietotekniikan yhdentymistä, mikä on odotettavissa. Tietämystekniikkaa ei pidä nähdä tälläkään hetkellä perinteistä atk-tekniikkaa korvaavana vaihtoehtona.

Mikrotietokonepohjaisia asiantuntijajärjestelmäkehittämiä on markkinoilla runsaasti. Kehittimien ominaisuudet ja laatu vaihtelevat ja oikean tuotteen löytäminen saattaa tuottaa

vaikeuksia. Valittaessa sopivaa kehitintä on pidettävä mielessä, että erilaisiin ongelmiin sopivat erilaiset kehittimet. Nykyiset mikropohjaiset kehittimet käyttävät yleensä sääntöjä tai kehyksiä tietämyksen esittämisessä. Hybridikehittimissä on yhdistetty eri tietämyksen esittämistekniikat, ja tällaisten kehittimien suosion odotetaan kasvavan.

Tietekniikassa on paljon suunnittelu- ja ylläpitotehtäviä, jotka vaativat tieinsinööriltä käytännön kokemusta. Asiantuntijajärjestelmät soveltuvat useisiin tämän tyyppisten ongelmien ratkaisuun. Rajattujen ja selväpiirteisten ongelmien ratkaisumenetelmät voidaan siirtää mikropohjaisiin asiantuntijajärjestelmiin. Yhdysvalloissa on jo tehty lukuisia mikropohjaisia tietekniikan asiantuntijajärjestelmiä, joista valtaosa on vielä prototyyppejä.

Tämän työn yhtenä osana rakennettiin mikrotietokonepohjainen murskausharjoitus asiantuntijajärjestelmä. Tämä asiantuntijajärjestelmä tehtiin maa- ja kalliorakennustöiden murskausharjoituksen pohjalta. Järjestelmälle annetaan lähtötietoina murskattavan louheen määrä ja raekoko sekä murskauksessa tehtävät lajitteet. Järjestelmä valitsee tietämyskannasta olevista murskaimista sopivat ja kyselee käyttäjältä murskaimiin liittyviä tietoja, kuten asetusta ja iskuliikettä. Järjestelmä laskee murskauksessa syntyvien lajitteiden määrät ja eri murskausvaiheisiin kuluvat ajat.



## ENGLISH SUMMARY

The development of expert systems and the tools for building them has been rapid. New products have appeared and their quality is constantly improving. Expert systems have proved to be efficient tools for solving small ill-structured problems.

A lot of research has been done by both universities and commercial enterprises on building more efficient systems. Knowledge acquisition is a very important part of building an expert system and it has been extensively studied. New methods for knowledge acquisition have been developed and new tools have also appeared. Most of them are based on induction. In the future, expert systems and other computer systems will probably integrate. Today conventional computer programs are used for large amounts of data and expert systems are used for small amounts of 'high level' data, knowledge and expertise. The future computer systems will be able to handle large amounts of intelligent data. Many of the expert system shells already have interfaces to other programs.

There are several microcomputer-based expert system shells commercially available. Choosing the right one from them is a difficult task because different shells are suitable for different tasks. Most of the microcomputer-based shells are either rule-based or frame-based. When choosing the right tool for the problem, it is important to check if the product has training, good documentation and support. New hybrid tools offer different knowledge representation techniques. Usually they use object-oriented programming techniques to represent elements of the problem as objects, which can contain rules.

The field of highway engineering is full of problems

requiring human expertise. Conventional computer programs cannot solve these problems because of their complexity. Expert systems have great potential for solving these difficult qualitative problems.

A small microcomputer-based expert system for assisting students in planning an aggregate crushing plant was developed as a part of this study. This demonstration prototype was developed with Xi Plus and it runs on an IBM-AT-compatible microcomputer. The system is rule-based and consists of three separate knowledge-bases. The user inputs data about the aggregate that is being crushed and the system chooses suitable crushers and calculates the time being used and the amount of different fractions that result from the crushing.

## LÄHDELUETTELO

- /1/ Barrett, M., VIA-yhtiöiden asiantuntijajärjestelmä seminaari "Recent developments in induction", Helsinki, 9.5.1989.
- /2/ Barrett, M., Beerel, A., Expert Systems in Business A Practical Approach, Ellis Horwood Limited, England 1988, 259 s.
- /3/ Cohn, L., Harris, R., Bowlby, W. Knowledge Acquisition for Domain Experts, Journal of Computing in Civil Engineering. Vol. 2, No. 2, April, 1988, ss. 107-120.
- /4/ Farghi, A., Joshua, S., Demetsky, M. Expert System for the Evaluation of the Rail/Highway Crossings., Journal of Computing in Civil Engineering. Vol. 2, No. 1, January, 1988, ss. 21-37.
- /5/ Goodall, A., The Guide to Expert Systems, Learned Information, Oxford 1985, 220 s.
- /6/ Hakkarainen, K., Insinööriyötä avustavat asiantuntijajärjestelmät, VTT, Tiedotteita 487, Espoo 1985, 44 s.
- /7/ Harmon, P., Maus, R., Morrissey, W., Expert Systems tools and applications, John Wiley & Sons, USA 1988, 289 s.
- /8/ Harris, R., Cohn, L., Bowlby, W. Designing Noise Barriers Using Expert System CHINA., Journal of Transportation Engineering. Vol.113, No. 2, March, 1988 ss. 127-138.
- /9/ Hartikainen, O-P., Kalliorakennustekniikka, Otakustantamo Espoo 1979, 205 s.
- /10/ Hayes-Roth F., Waterman, D., Lenat, D., Building Expert Systems, Addison-Wesley Publishing Company 1983, 444 s.



- /11/ Heikkilä, J., Maa-ainesten murskaus, Teknillinen Korkeakoulu, Espoo 1988, 246 s.
- /12/ Kautto-Koivula, K., Tietämysjärjestelmien arkkitehtuuri VTT, Tiedotteita 577, Espoo 1986, 101s.
- /13/ Koskela, L., Serén, K-J., Betoninvalinnan asiantuntijajärjestelmä, VTT, Tiedotteita 722, Espoo 1987, 62 s.
- /14/ Kähkönen, K., Tietämystekniikka tien rakenteen suunnittelussa, diplomityö TKK 1986, 66 s.
- /15/ Linderholm, O., 'Leave it to the experts', PC World February, 1987.
- /16/ Linderholm, O., Crystal & VP-Expert, PC World, April, 1987.
- /17/ Linnainmaa, S., VTT ATK-palvelutoimisto, Asiantuntijajärjestelmät-kurssi, 'Tietämystekniikan soveltuvuuden arviointi 7 s.
- /18/ Linnainmaa, S., VTT ATK-palvelutoimisto, Asiantuntijajärjestelmät-kurssi, 'Apuvälineiden valinta' 6 s.
- /19/ Martin, J., Oxman, S. : Building Expert Systems A Tutorial, Prentice Hall, USA 1988, 457 s.
- /20/ Rantanen, J., Tietämysjärjestelmien rakentaminen VTT, Tiedotteita 736, Espoo 1987, 144 s.
- /21/ Richie, S., Yeh, C., Mahoney, J., Jackson, N., Surface Condition Expert System for Pavement Rehabilitation Planning, Journal of Transportation Engineering Vol 113., No 2, March 1987, ss. 155-167.
- /22/ Rolston, D., W., Principles of Artificial Intelligence and Expert Systems Development, McGraw-Hill, Singapore 1988, 257 s.

- /23/ Salo, K., Asiantuntijajärjestelmäkehittimen valinta mikrotietokoneelle, Imatran Voima, T & K-tiedotteita, Helsinki 1988, 31 s.
- /24/ Shapiro, E., AI, AI, Oh !, Byte, June 1987, ss 321-323.
- /25/ Simons, G., L. Expert Systems and Micros, The National Computing Centre Limited, England, 1983, 247 s.
- /26/ Simons, G., L. Towards Fifth Generation Computers The National Computing Centre Limited, England, 1983, 226 s.
- /27/ Stammer, R., Abkowitch, M., Microcomputer Applications in Transportation II, American Society of Civil Engineering, USA 1987.
- /28/ Tarjanne, A., Tietokoneavusteinensuunnittelu tietekniikassa, asiantuntijajärjestelmät luento, Otaniemi 24.4.1989.
- /29/ Tarjanne, A., VIA-yhtiöiden asiantuntijajärjestelmä seminaari, Helsinki 9.5.1989.
- /30/ Tello, E., Acquaint application review, Byte, June 1987 ss. 265-272.
- /31/ Tello, E., Personal Consultant Plus review, Byte, October 1987, ss. 242-244.
- /32/ VP Expert, User Manual, Paperback Software, 1987.
- /33/ Xi Plus, User Manual Release 3, Expertech 1989.
- /34/ Xi Plus, Kurssimateriaali, Via Yhtiöt, 1989.